(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 417 489**

(51) Int. CI.:

G06F 17/30 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **10.04.2008 E 08737415 (3)**

(97) Fecha y número de publicación de la concesión europea: **03.04.2013 EP 2210194**

(54) Título: **Mecanismo y método de detección de colisión de Protocolo Ligero de Acceso a Directorios (LDAP)**

(30) Prioridad:

06.11.2007 US 985710 P

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
08.08.2013

(73) Titular/es:

TELEFONAKTIEBOLAGET LM ERICSSON (PUBL) (100.0%)
164 83 Stockholm , SE

(72) Inventor/es:

ALARCON ALONSO, ANTONIO;
BARTOLOME RODRIGO, MARIA CRUZ y
VEGA ARNAEZ, JULIO

(74) Agente/Representante:

DE ELZABURU MÁRQUEZ, Alberto

DESCRIPCIÓN

Mecanismo y método de detección de colisión de Protocolo Ligero de Acceso a Directorios (LDAP)

RIVINDICACIÓN DE SOLICITUD DE U.S. ANTERIORMENTE ENVIADA
Esta solicitud reclama la Solicitud de Patente Provisional de U.S. de Nº de Serie 60/985.710, que fue presentada el 6 de Noviembre de 2007.

CAMPO TÉCNICO
La presente invención se refiere a un mecanismo de detección de colisión de LDAP y a un método para permitir que un cliente de LDAP detecte y evite una colisión de operación de actualización en una entrada dentro de un directorio de LDAP.

ANTECEDENTES
Se definen aquí las siguientes abreviaturas, a algunas de las cuales se hace referencia en la siguiente descripción de la técnica anterior y de la presente invención.

AAA	Autenticación, Autorización y Contabilidad (Authentication, Authorization and Accounting, en inglés)
CAS	Sistema de Administración de Abonados (Customer Administration System, en inglés)
CDC	Contador de Detección de Colisiones (Collision Detection Counter, en inglés)
CS	Circuitos Conmutados (Circuit Switched, en inglés)
DIT	Árbol de Información de Directorios (Directory Information Tree, en inglés)
DN	Nombre Distinguido (Distinguished Name, en inglés)
FE	Ordenador Frontal (Front End, en inglés)
EMA	Multi Activación de Ericsson (Ericsson Multi Activation, en inglés)
GSM	Sistema Global para Comunicaciones mediante Telefonía Móvil (Global System for Mobile Communications, en inglés)
HLR	Registro de Ubicación Local (Home Location Register, en inglés)
HSS	Servidor de Abonados Local (Home Subscriber Server, en inglés)
IT	Tecnología de la Información (Information Technology, en inglés)
LDAP	Protocolo Ligero de Acceso a Directorios (Lightweight Directory Access Protocol, en inglés)
MAP	Parte de Aplicación de Móviles (Mobile Application Part, en inglés)
NDC	Número de Colisiones para ser Detectadas
PS	Paquetes Conmutados (Packet Switched, en inglés)

En el campo de las comunicaciones, la arquitectura de la base de datos de abonados actual está siendo retada por una arquitectura que está apareciendo de tecnología de IT de corriente principal donde los datos del abonado se guardan en una o más bases de datos separadas de los nodos que proporcionan el servicio específico. Este planteamiento se denomina una arquitectura de multi-capa y un dibujo de ejemplo que ilustra gráficamente este tipo de arquitectura que se utiliza en el campo de las comunicaciones de telecomunicaciones inalámbricas se muestra en la FIGURA 1.

En referencia a la FIGURA 1, se ilustra una red de comunicaciones 100 que tiene una red de IMS 101 que incluye un FE de HSS 102 y un FE de aprovisionamiento 104 y una red de núcleo 106 de CS/PS que incluye un FE de HLR/AuC 108 y un FE de AAA 110. El FE de HSS 102, el FE de aprovisionamiento 104, el FE de HLR/AuC 108 y el FE de AAA 110 se comunican todos ellos con una base de datos centralizada 112, que puede estar acoplada a un EMA 114 (por ejemplo, un dispositivo de aprovisionamiento de suscripción), que a su vez está acoplado a un CAS convencional 116. La red de IMS 101 y la red de núcleo 106 de CS/PS etc... incluyen más componentes que los mostrados en esta memoria, pero en aras de la claridad sólo los componentes que son relevantes para la presente explicación se han descrito en esta memoria.

Esta arquitectura de multi-capa proporciona varias ventajas, la no menor de las cuales es la menos costosa escalabilidad en la capa lógica de servicio o la capacidad para consolidar los datos del abonado de manera que la administración de los abonados es más fácil y menos costosa en comparación con las redes de comunicación mediante telefonía móvil tradicionales. En esta arquitectura de multi-capa, los nodos monolíticos tradicionales (incluyendo la lógica tanto de datos como de procesamiento) tales como el HSS, el HLR y el AAA han evolucionado para procesar ordenadores frontales (FE – Front Ends, en inglés) como el FE de HSS 102, el FE de aprovisionamiento 104, el FE de HLR/Auc 108 y el FE de AAA 110, mientras que los datos residen ahora en la base de datos centralizada 112 ó en una base de datos distribuida accesible para los ordenadores frontales anteriores, 102, 104, 108 y 110.

En esta arquitectura de multi-capa, el FE de HLR/AuC 108 (por ejemplo) tras recibir algún evento externo (es decir, un mensaje de MAP) desde la red de núcleo de CS/PS 106, tiene que leer datos relativos al abonado de la base de datos centralizada 112, procesar esos datos leídos a la vista de los datos recibidos de la red de núcleo de CS/PS

106, y dependiendo del resultado de este procesamiento interno puede desear modificar los datos relativos al abonado que están actualmente almacenados dentro de la base de datos centralizada 112. Una explicación detallada acerca de este proceso se proporciona a continuación con respecto a la FIGURA 2 (TÉCNICA ANTERIOR) donde la base de datos centralizada 112 es un directorio de LDAP 112 y el FE de HLR/AuC 108 es un cliente de LDAP 108.

En referencia a la FIGURA 2 (TÉCNICA ANTERIOR), hay un diagrama de flujo de señal que ilustra cómo un cliente de LDAP 108 tradicional lee datos 202 de una entrada a un directorio de LDAP 112 tradicional (o servidor de LDAP 112) y modifica a continuación los datos 202 que están almacenados en la entrada del directorio de LDAP 112 tradicional. Las etapas asociadas con dónde lee el cliente 108 y a continuación modifica los datos son como sigue:

1a-1b. Lee algunos datos 202 de una entrada (o entradas) en el directorio de LDAP 112 en las que el cliente 108 está interesado, con cualquier propósito. Esto requiere que el cliente 108 envíe una operación de Solicitud de Búsqueda de LDAP al directorio de LDAP 112 (etapa 1a). El servidor de LDAP 112 envía a continuación una copia de los datos 202 de la entrada (o entradas) utilizando una o más respuestas Respuesta a Búsqueda de LDAP (etapa 1b).

2. El cliente 108 puede utilizar lógica de aplicación para procesar los datos leídos 202 con cualquier propósito, como por ejemplo, extraer información, procesar los datos leídos frente a algún otro dato internos, enviar los datos leídos a otro nodo/proceso, imprimir algunos resultados, evaluar algunas condiciones…. En este caso, el cliente 108 ha actualizado los datos 202'.

3a-3b. El cliente 108 desea llevar a cabo algunas actualizaciones en los datos 202 guardados en la entrada (o entradas) leída (o leídas) previamente en el directorio de LDAP 112. Esto requiere que el cliente 108 envíe una operación de Modificar Solicitud de LDAP con los datos actualizados 202' al directorio de LDAP 112 (etapa 3a) (nota: una operación de Modificar Solicitud de LDAP sería necesaria para cada entrada al directorio cuya actualización se solicita). El directorio de LDAP 112 actualiza la entrada para tener los datos 202' y envía al cliente 108 un mensaje de éxito en una operación de Modificar Respuesta de LDAP (Resultado éxito) (etapa 3b). Una sola operación de Modificar de LDAP aplica sólo a la entrada de objetivo, pero puede contener tantas operaciones de modificación (añadir/borrar/reemplazar) como se desee en el conjunto de tipos de atributos que se guardan en esa entrada particular.

Por desgracia, si hay más de uno o concurrentes clientes de LDAP 102 y 108 (por ejemplo) que pueden interactuar con el directorio de LDAP 112, entonces puede darse una situación problemática, como se muestra en la FIGURA 3 (TÉCNICA ANTERIOR), en la que el cliente 102 (cliente 2) sobrescribe los datos 202 que fueron leídos previamente por el cliente de LDAP 108 (cliente 1), pero que no fueron todavía modificados por el cliente de LDAP 108 (cliente 1). Las etapas son como sigue:

1a. El cliente 1 solicita leer algunos datos 202 del directorio de LDAP 112. Este mensaje podría ser una solicitud de lectura de cualquier cantidad de datos, y cualquier BÚSQUEDA de LDAP estándar puede ser aplicable y ser utilizada para solicitar los datos 202.

1b. El cliente 1 recibe estos datos 202 solicitados del directorio de LDAP 112. Esto puede ser llevado a cabo por medio de uno o de varios mensajes de LDAP (Entrada de Resultado de Búsqueda de LDAP), incluyendo un mensaje para indicar que toda la información solicitada ha sido enviada (Resultado de Búsqueda de LDAP Realizada). En este momento, el cliente 1 puede tomarse algún tiempo para llevar a cabo alguna lógica interna, con cualquier propósito, como por ejemplo llevar a cabo alguna comprobación de consistencia de los datos leídos, conectarse a otro nodo para solicitar algún otro dato basándose en algo leído… En este caso, el cliente ha actualizado los datos 202'.

2a. El cliente 2 solicita la lectura de algunos datos 202 del directorio de LDAP 112. Este mensaje de lectura puede solicitar los mismos datos que antes, o parte de los datos leídos previamente, o cualquier otro dato dentro del directorio de LDAP 112. En este ejemplo, se asume que al menos una parte de los mismos datos 202 leídos en la etapa 1b es solicitada por el cliente 2.

2b. El cliente 2 recibe los datos 202 solicitados del servidor de LDAP 112. Esto puede ser llevado a cabo por medio de uno o de varios mensajes de LDAP (Entrada de Resultado de Búsqueda de LDAP), incluyendo un mensaje para indicar que toda la información solicitada ha sido enviada (Resultado de Búsqueda de LDAP Completada). A continuación, el cliente 2 lleva a cabo cualquier procesamiento y lógica, utilizando o no utilizando los datos leídos para tal propósito. En este caso, el cliente 2 ha actualizado los datos 202''.

3a. El cliente 2 solicita la modificación de algunos (al menos un atributo) o todos los datos 202 leídos previamente por el cliente 1.

3b. La solicitud de modificación del cliente 2 tiene éxito. La modificación fue llevada a cabo con éxito, porque el directorio de LDAP 112 no tiene ninguna razón/información para no permitir esta modificación. Desde este momento en adelante, los datos 202 leídos previamente por el cliente 1 se quedan obsoletos, porque al menos una parte de esos datos han sido sobrescritos por el cliente 2.

4a. El cliente 1 solicita la modificación de los datos 202 leídos previamente. En particular, el cliente 1 solicita la modificación de uno o más de los atributos de los datos 202 leídos previamente que han sido total o parcialmente sobrescritos por el cliente 2 durante la etapa 3b previa.

4b. La solicitud de modificación del cliente 1 es un éxito y los datos 202' están ahora almacenados en el directorio de LDAP 112. El directorio de LDAP 112 no tiene ninguna razón o información para no permitir esta modificación particular de los datos. No obstante, puede suceder que algunas modificaciones que el cliente 1 pidió, basadas en el estado de los datos en la etapa 1b, pueden no ser válidas. Como resultado, pueden aparecer algunas inconsistencias en los datos.

En este caso particular, puede verse cómo los datos leídos por el cliente 1 han sido modificados por el cliente 2 antes de que el cliente 1 proceda a actualizar estos datos. Esto no resulta deseable. Por ejemplo, si las actualizaciones de los datos llevadas a cabo por el cliente 1 son dependientes de que un estado de servicio sea "habilitado", entonces puede ocurrir que el cliente 2 haya modificado este estado de servicio para que sea "deshabilitado", lo que significa que las actualizaciones por parte del cliente 1 no deberían ser posibles. Esto puede finalizar en un fallo debido a actualizaciones erróneas de los datos. Así, si el cliente de LDAP (por ejemplo, el FE de HLR/AuC 108) lleva a cabo una Búsqueda de LDAP, procesa la respuesta de LDAP y a continuación envía modificaciones (Modificación de LDAP) al directorio de LDAP 112. Entonces, no hay manera a día de hoy de asegurar que una vez que se ha recibido el Modificar de LDAP, la misma condición es aun válida, puesto que algunas modificaciones podrían haber sido llevadas a cabo en el directorio de LDAP 112 por otro cliente de LDAP (por ejemplo, el FE de HSS 102) después del momento de que la Búsqueda de LDAP fue respondida en respuesta a la solicitud del primer cliente de LDAP. Esta situación puede provocar inconsistencias en los datos. De acuerdo con esto, existe y ha existido la necesidad de solucionar este inconveniente particular y otros inconvenientes que se solucionan mediante la presente invención.

COMPENDIO

En un aspecto, la presente invención proporciona un directorio y un método para detectar y evitar colisiones en una entrada al directorio mediante operaciones de actualización de más de un ordenador frontal de cliente (FE DE HSS, FE DE HLR). El directorio y método llevan a cabo las etapas de: (a) recibir en el directorio una solicitud de un ordenador frontal de cliente para que lea datos en una entrada al directorio; (b) asignar en el directorio al menos un valor de detección de colisión dado correspondiente al menos a un subconjunto de los datos en la entrada al directorio; (c) enviar hacia el ordenador frontal del cliente los datos solicitados junto con el al menos un valor de detección de colisión dado; (d) recibir en el directorio una solicitud del ordenador frontal del cliente de modificar el al menos un subconjunto de los datos en la entrada al directorio, donde la solicitud incluye al menos un valor de detección de colisión actualizado para cada uno de al menos un valor de detección de colisión dado; (e) determinar en el directorio si el al menos un valor de detección de colisión actualizado está de acuerdo con un valor de detección de colisión actual correspondiente o no; (f) enviar hacia el ordenador frontal del cliente la aceptación de la solicitud de modificar el al menos un subconjunto de los datos en la entrada al directorio, donde el al menos un valor de detección de colisión actualizado está de acuerdo con el valor de detección de colisión correspondiente, o si no el rechazo de la solicitud; y (g) donde la solicitud resulte aceptable, cambiar el valor de detección de colisión actual al siguiente valor. Este esquema resulta deseable, puesto que evita las inconsistencias en los datos y los fallos debidos a actualizaciones erróneas de datos.

En otro aspecto, la presente invención proporciona un ordenador frontal de cliente y un método para detectar y evitar colisiones en una entrada a un directorio mediante una operación de actualización desde el ordenador frontal del cliente (FE de HSS, FE de HLR). El ordenador frontal del cliente y el método llevan a cabo las etapas de: (a) enviar una solicitud desde un ordenador frontal de cliente para leer datos en una entrada al directorio; (b) obtener los datos solicitados en el ordenador frontal del cliente desde entrada al directorio, junto con al menos un valor de detección de colisión dado correspondiente al menos a un subconjunto de los datos; (c) procesar los datos obtenidos en el ordenador frontal de cliente; (d) enviar una solicitud desde el ordenador frontal del cliente para modificar el al menos un subconjunto de los datos en la entrada al directorio, donde la solicitud incluye al menos un valor de detección de colisión actualizado para cada uno de los al menos un valor de detección de colisión dados; y (e) recibir en el ordenador frontal del cliente una aceptación de la solicitud para modificar el al menos un subconjunto de los datos en la entrada al directorio, donde el al menos un valor de detección de colisión actualizado está de acuerdo con un valor de detección de colisión correspondiente en el directorio, o si no el rechazo de la solicitud. Este esquema resulta deseable puesto que evita inconsistencias de datos y fallos debidos a actualizaciones erróneas de datos.

En otro aspecto más, la presente invención proporciona un sistema y método para permitir que el ordenador frontal del cliente detecte y evite una colisión de operación de actualización en una entrada dentro de un directorio

mediante: (a) enviar una solicitud de iniciar una transacción hacia un directorio; (b) recibir una respuesta que incluye un identificador de transacción desde el directorio; (c) enviar una solicitud de lectura de datos en la entrada al directorio, donde la solicitud incluye también el identificador de transacción; (d) recibir una respuesta que incluye los datos leídos y el identificador de transacción desde el directorio, donde el directorio bloquea los datos dentro de la entrada como parte de una transacción en curso, de manera que ningún otro cliente puede modificar los datos bloqueados; (e) procesar los datos leídos; (f) enviar una solicitud de modificar los datos en la entrada al directorio, donde la solicitud incluye los datos modificados y el identificador de transacción, y donde el directorio lleva a cabo la solicitud de modificar los datos debido a la presencia del identificador de la transacción; y (g) enviar una solicitud de detener la transacción al directorio. Este esquema resulta deseable puesto que evita inconsistencias de datos y fallos debidos a actualizaciones erróneas de datos.

Aspectos adicionales de la invención se explicarán, en parte, en la descripción detallada, figuras y cualquier reivindicación que sigue, y la parte será deducida de la descripción detallada, o puede ser aprendida mediante la puesta en práctica de la invención. Resultará evidente que tanto la descripción general anterior como la siguiente descripción detallada son sólo a modo de ejemplo y de explicación y no restrictivas de la invención tal como se presenta.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

Una más completa comprensión de la presente invención puede obtenerse por referencia a la siguiente descripción detallada cuando se toma junto con los dibujos que se acompañan:

la FIGURA 1 (TÉCNICA ANTERIOR) es un diagrama de una red de comunicaciones que tiene múltiples clientes de LDAP que se comunican con una base de datos de LDAP que se utiliza para ayudar a explicar un problema de colisión que se resuelve mediante la presente invención;

la FIGURA 2 (TÉCNICA ANTERIOR) es un diagrama de flujo de señal que ilustra cómo un cliente de LDAP tradicional lee datos de una entrada a un directorio de LDAP tradicional y a continuación modifica los datos que están almacenados dentro del directorio de LDAP tradicional;

la FIGURA 3(TÉCNICA ANTERIOR) es un diagrama de flujo de señal que ilustra cómo ocurre un problema de colisión cuando un cliente de LDAP sobrescribe los datos que habían sido leídos previamente pero que no habían sido modificados todavía por otro cliente de LDAP;

la FIGURA 4 es un diagrama de una red de comunicaciones que tiene múltiples clientes de LDAP mejorados que se comunican con una base de datos de LDAP mejorada que se utiliza para ayudar a explicar cómo es resuelto el problema de colisión mediante la presente invención;

las FIGURAS 5-6 son diagramas de secuencia de señal que se utilizan para ayudar a explicar cómo se resuelve el problema de colisión mediante un mecanismo/método de detección de colisión de acuerdo con una primera realización de la presente invención;

la FIGURA 7 es un diagrama de secuencia de señal utilizado para ayudar a explicar cómo se resuelve el problema de colisión mediante un mecanismo/método de acuerdo con una segunda realización de la presente invención;

la FIGURA 8 es un diagrama de secuencia de señal utilizado para ayudar a explicar cómo se resuelve el problema de colisión mediante un mecanismo/método de detección de colisión de acuerdo con una tercera realización de la presente invención; y

la FIGURA 9 es un diagrama de secuencia de señal utilizado para ayudar a explicar cómo se resuelve el problema de colisión mediante un mecanismo/método de detección de colisión de acuerdo con una cuarta realización de la presente invención.

DESCRIPCIÓN DETALLADA

En referencia a la FIGURA 4, hay un diagrama que ilustra una red de comunicaciones 400 de ejemplo que se utiliza para ayudar a explicar cómo los clientes de LDAP 402, 404, 408 y 410 y una base de datos de abonados de LDAP 412 pueden ser mejorados de acuerdo con la presente invención de manera que cualquiera de de los clientes de LDAP 402, 404, 408 y 410 mejorados puede detectar colisiones de operaciones de actualización en una entrada al directorio dentro de la base de datos de LDAP 412 (directorio de LDAP 412) mejorada. Como se muestra, la red de comunicaciones 400 tiene una red de IMS 401 que incluye clientes de LDAP 402 y 404 mejorados respectivamente representados como un FE de HSS 402 mejorado y un FE de aprovisionamiento 404 mejorado. La red de comunicaciones 400 también tiene una red de núcleo de CS/PS 406 que incluye clientes de LDAP 408 y 410 mejorados respectivamente representados como un FE de HLR/AuC 408 mejorado y un FE de AAA 410 mejorado. El FE de HSS 402 mejorado, el FE de aprovisionamiento 404 mejorado, el FE de HLR/AuC 408 mejorado y el FE de AAA 410 mejorado se comunican con la base de datos de LDAP centralizada 412 mejorada (o con una base de datos distribuida accesible para los ordenadores frontales 402, 404, 408 y 410 anteriores) que está acoplada a un EMA 414 (por ejemplo, un dispositivo de aprovisionamiento de suscripción) que a su vez está acoplado a un CAS 416. La red de IMS 401 y la red de núcleo de CS/PS 406 etc.… incluyen más componentes que los mostrados en esta memoria, pero en aras de la claridad sólo los componentes que son relevantes para la presente explicación han sido descritos en esta memoria.

Como se ha indicado anteriormente, la presente invención se refiere a un mecanismo/método de detección de colisión que permite que un cliente de LDAP 404 (por ejemplo) detecte una colisión de operación de actualización creada por un cliente de LDAP 408 diferente (por ejemplo) en una entrada dentro de un directorio de LDAP 412. En realidad, se describen en esta memoria cuatro realizaciones diferentes de tal mecanismo/método de detección de colisión, permitiendo todas ellas que un cliente de LDAP detecte una colisión de operación de actualización creada por un cliente de LDAP diferente en una entrada dentro de un directorio de LDAP. La primera realización del mecanismo de detección de colisión se describe a continuación con la ayuda de dos diagramas de flujo de señal mostrados en las FIGURAS 5-6.

En la primera realización, el mecanismo de detección de colisión se basa en definir un nuevo tipo de atributo ENTERO de múltiples valores (denominado en esta memoria el “Contador de Detección de Colisión” (CDC – Collision Detection Counter, en inglés) en cada entrada del directorio de LDAP 412, donde las colisiones entre los clientes de LDAP 402, 404, 408 y 410 concurrentes desean ser detectadas. Por ejemplo, este nuevo atributo de CDC puede definirse como:

(<OID_asignado>
NOMBRE ’cdc’
DESC ‘Contador de Detección de Colisión’
SINTAXIS’1.3.6.1.4.1.1466.115.121.1.27’
X-ORIGEN ‘Mecanismo-CDC-Mutex’)

Para detalles acerca de cómo pueden definirse los atributos en el primer lugar, se hace referencia al RFC 4512 titulado “Lightweight Directory Access Protocol (LDAP): Directory Information Models”, de fecha de Junio de 2006.

Este nuevo atributo de CDC puede ser declarado como un atributo “mandatorio” en la clase o clases de objeto de estructura deseada o deseadas (es decir, la clase o clases de objeto utilizada o utilizadas para entrada o entradas al directorio). En este caso, cada entrada al directorio pertenecería a una y sólo una de las clases de objeto de estructura, y todos los atributos que se definen como mandatorios en la clase de objeto de estructura seleccionada serían aprovisionados cuando se crea la respectiva entrada al directorio. Alternativamente, el nuevo atributo puede ser definido en una nueva clase de objeto “auxiliar”. En este caso, una entrada pertenecería a cero o más clases de objeto auxiliares.

El CDC puede ser gestionado como un contador cíclico, así que puede ser actualizado mediante la siguiente expresión matemática (‘%’ representa el operador “módulo”):

$$CDC\ [n+1] = (CDC\ [n] + 1)\ \%\ M$$

donde “M” es el máximo número que este contador puede alcanzar (así que el CDC serán las siguientes secuencias: 0 → 1 → 2 … → (M-2) → (M-1) → 0 → 1 …). En la creación de la entrada al directorio no se requiere que el valor de CDC sea inicializado, pero se requeriría que fuese aprovisionado en la creación de la entrada.

Además, el mecanismo de detección de colisión se basa también en definir un nuevo parámetro (denominado en esta memoria NCD “Número de Colisiones para Detectar”) que indica el máximo número de clientes de LDAP 402, 404, 408 y 410 concurrentes que deben ser consideradas cuando se evitan colisiones entre ellas cuando se modifican datos en el directorio de LDAP 412.

Además, el mecanismo de detección de colisión se basa en añadir algunas operaciones de modificación “extra” (para el valor de atributo de CDC) a una operación de Solicitud de Modificar de LDAP estándar. Un aspecto clave de esta modificación particular implica utilizar un comportamiento de LDAP estándar, donde un mensaje de Modificar de LDAP es rechazado en caso de que intente “añadir” un valor a un atributo que ya existe y que tiene el mismo valor. Esto se explica con más detalle a continuación.

Utilizando estas definiciones y el mensaje de Modificar de LDAP modificado, los clientes de LDAP 402, 404, 408 y 410 seguirían estas reglas:

1. Leer el valor del atributo de CDC actual que está en los datos obtenidos (almacenados en una entrada al directorio de LDAP 412) que había sido preparado para la detección de colisión. En realidad, podría leerse cualquier otro número de valores de atributos de CDC, desde la misma entrada o desde cualquier otra entrada al directorio de LDAP 412.

2. Cuando se modifica cualquier dato en la entrada que ha sido preparada para la detección de colisión, deben tenerse en cuenta dos cosas:

A. El mensaje de Modificar de LDAP puede incluir la operación u operaciones de “añadir” para el atributo de CDC, para los siguientes valores:

CDC = (Leer valor de CDC + 1) % M

CDC = (Leer valor de CDC + 2) % M

…

CDC = (Leer valor de CDC + NCD) % M

Nota: dependiendo del tipo de atributo de CDC puede ser necesario considerar el módulo M, en caso de que se utilice un contador cíclico.

B. Tras la operación u operaciones de “añadir”, el mensaje de modificar de LDAP incluye una operación de “reemplazar” como sigue:

CDC = Leer valor de CDC + 1

Y, a continuación hay un ejemplo para ayudar a explicar mejor el uso de estas reglas:

Asúmase que un valor de CDC leído por un cliente de LDAP 402 (por ejemplo) es 0, para una entrada que ha sido preparada para la detección de colisión, al mismo tiempo que los datos son leídos desde el directorio de LDAP 412. Además, asúmase que el CDC está implementado como un contador cíclico, y que M (valor máximo) es 65536. Entonces, el atributo de CDC puede almacenar valores de 0 a 65535 como sigue:

$$0 \rightarrow 1 \rightarrow 2 \rightarrow \ldots \rightarrow 65534 \rightarrow 65535 \rightarrow 0 \rightarrow 1$$

También, asúmase que el número de colisiones para ser detectadas es puesto a 4 (NCD = 4). Esto significa que cada cliente de LDAP (por ejemplo, el cliente de LDAP 402) es capaz de “detectar” si 1, 2, 3 ó incluso 4 otros clientes de LDAP (por ejemplo, los clientes de LDAP 404, 408 y 410) han actualizado la entrada “protegida” en el tiempo desde el momento en que el CDC fue leído inicialmente y el momento en que el primer cliente de LDAP (por ejemplo, el cliente de LDAP 402) decide solicitar alguna actualización para los datos en la entrada.

A continuación, el primer cliente de LDAP (por ejemplo, el cliente de LDAP 402) envía un mensaje de Modificar de LDAP que incluye la siguiente operación o las siguientes operaciones de “añadir”:

CDC = 1
CDC = 2
CDC = 3
CDC = 4

Además, el mensaje de Modificar de LDAP después de las operaciones de “añadir” tiene una operación de “reemplazar” como sigue:

CDC = 1

Cómo responde el directorio de LDAP 412 a este mensaje de Modificar de LDAP que permite la detección de una colisión en un acceso concurrente y si hay una colisión entonces evitar que se lleven a cabo las actualizaciones solicitadas se explica con detalle con respecto a dos diagramas de secuencia de ejemplo. Para completar más, los dos diagramas de secuencia de ejemplo incluyen también una descripción acerca de las etapas en las que los clientes de LDAP leen los datos, procesan los datos leídos y a continuación envían una solicitud de modificar los datos dentro del directorio de LDAP 412.

En referencia a la FIGURA 5, hay un diagrama de secuencia de señal que ilustra un ejemplo de implementar el mecanismo de detección de colisión donde el valor de NCD seleccionado fuese igual a lo que se necesita para detectar adecuadamente una colisión de acuerdo con la presente invención. En este ejemplo, se muestran tres clientes de LDAP 402, 404 y 408 concurrentes (Clientes 1, 2 y 3) dado que tal NCD = 3 sería suficiente, pero más NDCs como NDC = 4 sería también suficiente para evitar que se lleve a cabo una actualización solicitada por uno de los clientes de LDAP 402, 404 y 408 si hubiese una colisión entre cualquiera de los clientes de LDAP 402, 404 y 408. Las etapas son como sigue:

1a. El cliente 1 envía un mensaje para leer algunos datos 502 de una entrada al directorio de LDAP 412. Este mensaje sería una solicitud para leer cualquier cantidad de datos incluyendo el CDC, y cualquier BÚSQUEDA DE LDAP estándar puede ser utilizada para solicitar los datos 502 y el valor de CDC.

1b1. – 1b2. El cliente 1 recibe los datos solicitados 502 y el valor de CDC desde el directorio de LDAP 412. Esto puede ser realizado por medio de una Entrada de Resultado de Búsqueda de LDAP que incluye la información solicitada (DN, lista de atributos, valor de CDC) (etapa 1b1) y por medio de un Resultado de Búsqueda de LDAP Completada que indica que la búsqueda fue un éxito (etapa 1b2). En este ejemplo, la copia local del cliente 1 de los datos leídos tiene un CDC = (x). En este momento, el cliente 1 puede tomarse un tiempo para ejecutar alguna lógica interna, con cualquier propósito, como por ejemplo realizar alguna comprobación de consistencia en los datos leídos, conectarse a otro nodo para solicitar algún otro dato basándose en algo leído. En este caso, el cliente 1 ha actualizado los datos 502'.

2a. El cliente 2 envía un mensaje para leer al menos una porción de los mismos datos en la misma entrada al directorio de LDAP 412. Este mensaje sería una solicitud para leer cualquier cantidad de datos incluyendo el CDC, y cualquier BÚSQUEDA de LDAP estándar puede ser utilizada para solicitar los datos y el valor de CDC.

2b1. – 2b2. El cliente 2 recibe los datos 502 solicitados y el valor de CDC desde el directorio de LDAP 412. Esto puede ser realizado por medio de una Entrada de Resultado de Búsqueda de LDAP que incluye la información solicitada (DN, lista de atributos, valor de CDC) (etapa 2b1) y mediante un Resultado de Búsqueda de LDAP Completada, que indica que la búsqueda fue un éxito (etapa 2b2). En este ejemplo, la copia local del cliente 2 de los datos leídos tiene un CDC = (x). En este momento, el cliente 2 puede tomarse un tiempo para ejecutar alguna lógica interna, con cualquier propósito, como por ejemplo, llevar a cabo alguna comprobación de consistencia en los datos leídos, conectarse a otro nodo para solicitar algún otro dato basándose en algo leído…. En este caso, el cliente 2 ha actualizado los datos 502''.

3a. El cliente 2 solicita la modificación de los datos leídos 502, al menos una porción de los cuales también fue leída por el cliente 1. En particular, el cliente 2 envía una Solicitud de Modificar de LDAP que incluye: (1) un objeto de base: entrada de DN; (2) tres operaciones de añadir: Añadir: CDC = (x) + 1; CDC = (x) + 2; CDC = (x) + 3; (3) los datos actualizados 502''; y (4) una operación de reemplazar: Reemplazar: CDC = (x) + 1.

3b. El cliente 2 recibe una Respuesta a Modificar de LDAP desde el directorio de LDAP 412 indicando que la solicitud de modificación fue un éxito. La modificación fue correctamente llevada a cabo, porque el directorio de LDAP 412 no tenía un atributo de CDC que tuviese el mismo valor que estaba en alguna de las tres operaciones de "añadir". En particular, el directorio de LDAP 412 tenía un valor de CDC de (x) antes de las tres operaciones de añadir y esas operaciones de añadir fueron para CDC = (x) + 1, CDC = (x) + 2, CDC = (x) + 3. RECUÉRDESE: el comportamiento del LDAP estándar es tal que un mensaje de Modificar de LDAP sería rechazado si intentase "añadir" un valor a un atributo que ya existe y que tiene el mismo valor (explicado con detalle en la etapa 6a). En este punto, el directorio de LDAP 412 ha asociado la entrada de leer con los datos 502'' y un CDC = (x) + 1.

4a. El cliente 3 envía un mensaje para leer al menos una porción de los mismos datos de la misma entrada al directorio de LDAP 412. Este mensaje sería una solicitud de leer cualquier cantidad de datos incluyendo el CDC, y cualquier BÚSQUEDA de LDAP estándar puede ser utilizada para solicitar los datos y el valor de CDC.

4b1. – 4b2. El cliente 3 recibe los datos solicitados 502'' desde el directorio de LDAP 412. Esto puede ser realizado por medio de una Entrada de Resultado de Búsqueda de LDAP que incluye la información solicitada (DN, lista de atributos, valor de CDC) (etapa 4b1) y por medio de un Resultado de Búsqueda de LDAP Completada que indica que la búsqueda tuvo éxito (etapa 4b2). En este ejemplo, la copia local del cliente 3 de los datos leídos tiene un CDC = (x) + 1. En este momento, el cliente 3 puede tomarse un tiempo para ejecutar alguna lógica interna, con cualquier propósito, como por ejemplo, llevar a cabo algunas comprobaciones de consistencia en los datos leídos, conectarse a otro nodo para solicitar algún otro dato basándose en algo leído…. En este caso, el cliente 3 había actualizado los datos 502'''.

5a. El cliente 3 solicita la modificación de los datos leídos, al menos una porción de los cuales también fue leída desde la misma entrada por los clientes 1 y 2. En particular, el cliente 3 envía una Solicitud de Modificar de LDAP que incluye: (1) un objeto de base: Entrada de DN; (2) tres operaciones de añadir: Añadir: CDC = (x) + 1; CDC = (x) + 2; CDC = (x) + 3; (3) los datos modificados 502'''; y (4) una operación de reemplazar: Reemplazar: CDC = (x+1) + 1.

5b. El cliente 3 recibe una Respuesta a Modificar de LDAP desde el directorio de LDAP 412 indicando que la solicitud de modificación tuvo éxito. La modificación fue llevada a cabo con éxito, porque el directorio de LDAP 412 no tenía un atributo de CDC que tuviese el mismo valor que estaba en las tres operaciones de "añadir". En particular, el directorio de LDAP 412 tenía un valor de CDC de (x+1) antes de las tres operaciones de añadir y esas operaciones de añadir eran para CDC = (x+1) + 1; CDC = (x+1) + 2; CDC =

(x+1) + 3. RECUÉRDESE: el comportamiento dl LDAP estándar es tal que un mensaje de Modificar de LDAP sería rechazado si tratase de “añadir” un valor a un atributo que ya existe y que tiene el mismo valor (explicado con detalle en la etapa 6a). En este punto, el directorio de LDAP 412 ha asociado la entrada leída con los datos 502’’’ y un CDC = (x) + 2.

6a. El cliente 2 solicita la modificación de los datos 502 que fueron leídos de nuevo en las etapas 1a-1b. En particular, el cliente 1 envía una Solicitud de Modificar de LDAP que incluye: (1) un objeto de base: entrada de DN; (2) tres operaciones de añadir: Añadir: CDC = (x) + 1; CDC = (x) + 2; CDC = (x) + 3; (3) datos 502’ modificados; y (4) una operación de reemplazar: Reemplazar: CDC = (x) + 1.

6b. El cliente 2 recibe una Respuesta a Modificar de LDAP desde el directorio de LDAP 412 indicando que la solicitud de modificación no tuvo éxito. La modificación no fue correctamente llevada a cabo, porque el directorio de LDAP 412 tenía un atributo de CDC, a saber, CDC = (x) + 2 que es el mismo valor que estaba en la segunda operación de “añadir”. RECUÉRDESE: el comportamiento del LDAP estándar es tal que un mensaje de Modificar de LDAP sería rechazado si intentase “añadir” un valor a un atributo que ya existe y que tiene ese mismo valor. Esto ocurrió en la etapa 6a, puesto que tal colisión fue detectada con éxito.

En referencia a la FIGURA 6, hay un diagrama de secuencia de señal que se proporciona para ilustrar un problema que puede ocurrir si el valor de NCD seleccionado es menor que el necesario, lo que hace difícil detectar adecuadamente todas las posibles colisiones de acuerdo con la presente invención. En este ejemplo, se muestran tres clientes de LDAP 402, 404 y 408 (clientes 1, 2 y 3) pero el NDC = 1, lo que puede provocar problemas en la detección de colisiones entre los clientes de LDAP 402, 404 y 408. Las etapas son como sigue:

1a. El cliente 1 envía un mensaje de leer algunos datos 502 de una entrada al directorio de LDAP 412. Este mensaje sería una solicitud de leer cualquier cantidad de datos incluyendo el CDC, y cualquier BÚSQUEDA de LDAP estándar puede ser utilizada para solicitar los datos 502 y el valor de CDC.

1b1. – 1b2. El cliente 1 recibe los datos 502 solicitados y el valor de CDC del directorio de LDAP 412. Esto puede ser realizado por medio de una Entrada de Resultado de Búsqueda de LDAP que incluye la información solicitada (DN, lista de atributos, valor de CDC) (etapa 1b1) y por medio de un Resultado de Búsqueda de LDAP Completada que indica que la búsqueda tuvo éxito (etapa 1b2). En este ejemplo, la copia local del cliente 1 de los datos leídos tiene un CDC = (x). En este momento, el cliente 1 puede tomarse un tiempo para ejecutar alguna lógica interna, con cualquier propósito, como por ejemplo llevar a cabo algunas comprobaciones de consistencia de los datos leídos, conectarse a otro nodo para solicitar algún otro dato basándose en algo leído… En este caso, el cliente 1 ha actualizado los datos 502’.

2a. El cliente 2 envía un mensaje para leer al menos una porción de los mismos datos de la misma entrada al directorio de LDAP 412. Este mensaje sería una solicitud de leer cualquier cantidad de datos incluyendo el CDC, y cualquier BÚSQUEDA de LDAP estándar puede ser utilizada para solicitar los datos y el valor de CDC.

2b1 – 2b2. El cliente 2 recibe los datos 502 solicitados y el valor de CDC del directorio de LDAP 412. Esto puede ser realizado por medio de una Entrada de Resultado de Búsqueda de LDAP que incluye la información solicitada (DN, lista de atributos, valor de CDC) (etapa 2b1) y por medio de un Resultado de Búsqueda de LDAP Completada, que indica que la búsqueda tuvo éxito (etapa 2b2). En este ejemplo, la copia local de los datos leídos del cliente 2 tiene un CDC = (x). En este momento, el cliente 2 se toma un tiempo para ejecutar alguna lógica interna, con cualquier propósito, como por ejemplo realizar algunas comprobaciones de consistencia sobre los datos leídos, conectarse a otro nodo para solicitar algún otro dato basándose en algo leído… En este caso, el cliente 2 ha actualizado los datos 502’’.

3a. El cliente 2 solicita la modificación de los datos 502 leídos, al menos una porción de los cuales también fue leída por el cliente 1. En particular, el cliente 2 envía una Solicitud de Modificar de LDAP que incluye: (1) un objeto de base: entrada de DN; (2) una operación de añadir: Añadir: CDC = (x) + 1; (3) datos modificados 402’’; y (4) una operación de reemplazar: Reemplazar: CDC = (x) + 1.

3b. El cliente 2 recibe una Respuesta a Modificar de LDAP desde el directorio de LDAP 412 indicando que la solicitud de modificación tuvo éxito. La modificación fue llevada a cabo con éxito, porque el directorio de LDAP 412 no tenía un atributo de CDC que tuviese el mismo valor que había en la una operación de “añadir”. En particular, el directorio de LDAP 412 tenía un valor de CDC de (x) antes de la operación de añadir de CDC = (x) + 1. RECUÉRDESE: el comportamiento de LDAP estándar es tal que un mensaje de modificación de LDAP sería rechazado si intentase “añadir” un valor a un atributo que ya existe y que tiene el mismo valor. En este punto, el directorio de LDAP 412 ha asociado la entrada de leer con los datos 402’’ y un CDC = (x) + 1.

4a. El cliente 3 envía un mensaje de leer al menos una porción de los mismos datos de la misma entrada al directorio de LDAP 412. Este mensaje sería una solicitud de leer cualquier cantidad de datos incluyendo el CDC, y cualquier BÚSQUEDA de LDAP estándar puede ser utilizada para solicitar los datos y el valor de CDC.

4b1. – 4b2. El cliente 3 recibe los datos 502'' solicitados del directorio de LDAP 412. Esto puede ser realizado por medio de una Entrada de Resultado de Búsqueda de LDAP que incluye la información solicitada (DN, lista de atributos, valor de CDC) (etapa 4b1) y por medio de un Resultado de Búsqueda de LDAP Completada que indica que la búsqueda fue un éxito (etapa 4b2). En este ejemplo, la copia local del cliente 3 de los datos leídos tiene un CDC = (x) + 1. En este momento, el cliente 3 puede tomarse algún tiempo para ejecutar alguna lógica interna, con cualquier propósito, como por ejemplo, llevar a cabo algunas comprobaciones sobre la consistencia de los datos leídos, conectase a otro nodo para solicitar algún otro dato basándose en algo leído… En este caso, el cliente 3 ha actualizado los datos 502'''.

5a. El cliente 3 solicita la modificación de los datos leídos, al menos una porción de los cuales también fue leída desde la misma entrada por los clientes 1 y 2. En particular, el cliente 3 envía una Solicitud de Modificar de LDAP que incluye: (1) un objeto de base: entrada de DN; (2) una operación de añadir: Añadir: CDC = (x+1) + 1; (3) los datos 502''' modificados; y (4) una operación de reemplazar: Reemplazar: CDC = (x+1) + 1.

5b. El cliente 3 recibe una Respuesta a Modificar de LDAP desde el directorio de LDAP 412 indicando que la solicitud de modificación tuvo éxito. La modificación fue llevada a cabo con éxito, porque el directorio de LDAP 412 no tenía un atributo de CDC que tuviese el mismo valor que había en la una operación de "añadir". En particular, el directorio de LDAP 412 tenía un valor de CDC de (x+1) antes de la operación de añadir CDC = (x+1) + 1. RECUÉRDESE: el comportamiento del LDAP estándar es tal que un mensaje de Modificar de LDAP sería rechazado si intentase "añadir" un valor a un atributo que ya existe y que tiene ese mismo valor. En este punto, el directorio de LDAP 412 ha asociado la entrada de leer con los datos 502''' y un CDC = (x) + 2.

6a. El cliente 1 solicita la modificación de los datos 502 que fueron leídos de nuevo en las etapas 1a-1b. En particular, el cliente 1 envía una Solicitud de Modificar de LDAP que incluye: (1) un objeto de base; entrada de DN; (2) una operación de añadir: Añadir: CDC = (x) + 1; (3) los datos 502' modificados; y (4) una operación de reemplazar: Reemplazar: CDC = (x) + 1.

6b. El cliente 1 recibe una Respuesta a Modificar de LDAP desde el directorio de LDAP 412 indicando que la solicitud de modificación tuvo éxito. La modificación fue llevada a cabo porque el directorio de LDAP 412 no tenía ningún atributo de CDC que tuviese el mismo valor que había en la una operación de "añadir". En particular, el directorio de LDAP 412 tenía un valor de CDC de (x) + 2 antes de la operación de añadir CDC = (x) + 1. Esto no resulta deseable puesto que ambos clientes 2 y 3 previamente modificaron los datos; no obstante, el mecanismo de detección de colisión de la presente invención no detectó estas colisiones porque el valor de NCD era demasiado bajo, lo que significaba que no había suficientes operaciones de añadir en los mensajes de Solicitud de Modificar de LDAP enviados por los clientes 1, 2 y 3.

Este diagrama de secuencia de señal particular muestra la necesidad de seleccionar el valor correcto para el parámetro NCD de manera que el mecanismo de detección de colisión pueda detectar todas las posibles colisiones. En este ejemplo, el NCD ha sido puesto en "1", así que no más de dos accesos concurrentes podrían haber sido detectados y modificaciones no deseadas de los datos no podrían haber sido evitadas. Así, para ayudar a asegurar la adecuada implementación del mecanismo de detección de colisión deben seguirse las siguientes dos recomendaciones: (1) el valor de "NCD": debería seleccionarse para ser mayor o igual que el máximo número de "clientes concurrentes" (es decir, los clientes de LDAP que acceden a la misma entrada en el mismo intervalo de tiempo) que podrían modificar esa entrada en el intervalo de tiempo después de que cualquiera de los clientes de LDAP obtuviese los datos (obtenidos con una operación de búsqueda de LDAP, incluyendo siempre el tipo de atributo de CDC) y el momento en que ese cliente de LDAP particular envía las actualizaciones para llevar a cabo (solicitado con una operación de Modificar de LDAP); y (2) seleccionar el valor de "M": donde el "M" debería ser seleccionado para ser mucho mayor que el valor de NCD.

En resumen, el mecanismo de detección de colisión asociado con la primera realización de la presente invención se basa en parte en el comportamiento estándar asociado con la operación de Modificar de LDAP que se describe en el documento anteriormente mencionado RFC 4511. Primero, es comportamiento estándar en el que toda una lista de modificaciones debe ser llevada a cabo dentro de la Solicitud de Modificar de LDAP con el fin de que estén listadas en una sola operación atómica (véase la sección 4.6 "Operación de Modificar" en RFC 4511). De esta manera, el mecanismo de detección de colisión de la presente invención tiene una Solicitud de Modificar de LDAP en la cual la operación o las operaciones de "añadir" se presentan siempre de manera que son llevadas a cabo antes de la operación de "reemplazar" (véanse las etapas 3a, 5a y 6a en la FIGURA 5).

Segundo, es comportamiento estándar el que añadir valores que están listados al atributo de modificación, resulte en la creación del atributo si es necesario. De esta manera, en el mecanismo de detección de colisión cuando se lleva a cabo un “añadir” en un atributo de múltiples valores, el nuevo valor solicitado es añadido a la lista existente. Si el valor ya existe, entonces se devuelve un código 20 (“Atributo O Valor Existe”) de acuerdo con el comportamiento estándar. Así, el mecanismo de detección de colisión utiliza el comportamiento estándar cuando establece la operación o las operaciones de añadir de manera que si el valor que va a ser “añadido” ya existe, entonces la modificación solicitada es rechazada. Este rechazo ocurre sólo si algún otro cliente ha actualizado el CDC a un nuevo valor (véase la etapa 6a en la FIGURA 5).

Tercero, es comportamiento estándar el que una operación de reemplazar resulte en el reemplazo de todos los valores existentes del atributo de modificación con el nuevo valor listado, y la creación del atributo si no existía ya. Un reemplazo sin ningún valor borrará todo el atributo si existe, y es ignorado si el atributo no existe. En el mecanismo de detección de colisión, un cliente utiliza la operación de “reemplazar” en el mensaje de Modificar de LDAP para actualizar el atributo de CDC al siguiente valor. Entonces, cuando el siguiente cliente que desea modificar los datos asumiendo que no se había detectado ninguna colisión con la operación o las operaciones de “añadir”, tendría la responsabilidad de actualizar el CDC al siguiente valor, así que otro cliente podría detectar esta modificación como una potencial colisión.

Como puede verse, debido al requisito del comportamiento estándar de atomicidad en la aplicación de esta lista de modificaciones en una Solicitud de Modificar de LDAP, el cliente 402, 404 y 408 puede esperar que no se lleve a cabo ninguna modificación del DIT en el directorio de LDAP 412 si la Respuesta a Modificar recibida indica algún tipo de error, y que todas las modificaciones solicitadas hayan sido llevadas a cabo si la Respuesta a Modificar indica que se ha completado con éxito la operación de Modificar.

En el ejemplo y explicaciones anteriores, se consideró que había un atributo de CDC para cada entrada (en el directorio de LDAP 412) que fue preparado para detectar colisiones. No obstante, puede suceder que la misma entrada pudiese tener múltiples atributos de CDC, para permitir una mayor granularidad en el mecanismo de detección de colisión; entonces, una colisión podría ser detectada para un grupo de atributos dentro de una entrada. Esto podría mejorar el mecanismo de detección de colisión puesto que permite un acceso concurrente más efectivo, siempre que no sean datos de acceso dentro de la misma entrada que pertenece al mismo grupo (es decir, al mismo CDC).

En referencia a la FIGURA 7, hay un diagrama de secuencia de señal utilizado para ayudar a explicar otro mecanismo de detección de colisión de ejemplo que puede ser utilizado para detectar una colisión. Este mecanismo de detección de colisión hace uso de lo que se conoce como capacidad de transacción en LDAP, donde se entiende que una transacción de LDAP está asociada con un grupo de una o varias operaciones en el cual se especifica un inicio y un fin para el grupo de operaciones. Además, cuando se envía una transacción de LDAP a un directorio de LDAP 412 hay una garantía de atomicidad donde bien todas las operaciones incluidas en tal transacción o ninguna de las operaciones son llevadas a cabo por el directorio de LDAP 412. En este sentido, puede verse desde la perspectiva de un cliente de LDAP 402, 404, 408 y 410 como si todas las operaciones incluidas en una transacción de LDAP pudiesen ser consideradas como una única operación. La segunda realización del mecanismo de detección de colisión de la presente invención utiliza la capacidad de transacción de LDAP para ayudar con la detección de colisión y evitar la colisión o colisiones. Cómo se lleva esto a cabo se describe en las etapas que siguen:

1a. El cliente 1 inicia una transacción en LDAP enviando una Solicitud de Iniciar Transacción al directorio de LDAP 412. Éste podría ser un nuevo mensaje de LDAP, que puede estar vacío, cuya función es solicitar que el directorio de LDAP 412 genere un identificador de transacción.

1b. El cliente 1 recibe una respuesta (Respuesta a Iniciar Transacción) a su solicitud previa para iniciar una transacción en LDAP. Esta respuesta podría ser un nuevo mensaje de LDAP que incluye, al menos, un identificador de transacción (TransId1). Este identificador se incluirá en cualquier mensaje de LDAP que el cliente 1 de LDAP requiere que sea procesado como parte de la misma transacción, es decir, que se considera parte de un grupo de mensajes, que a partir de la aplicación podrían ser considerados como una sola operación.

2a. El cliente 1 envía una solicitud (Solicitar Búsqueda de LDAP) de leer algunos datos 702 del directorio de LDAP 412. Este mensaje de BÚSQUEDA de LDAP incluye el identificador de transacción (TransId1) que acaba de ser generado, lo que significa que este mensaje debería ser parte de una transacción, es decir, de un grupo de mensajes. Este mensaje de BÚSQUEDA de LDAP podría solicitar la lectura de cualquier cantidad de datos 702, y cualquier BÚSQUEDA de LDAP estándar puede ser aplicable.

En este momento, el directorio de LDAP 412 implementa un mecanismo interno que es capaz, si es necesario, de bloquear “leer” datos (o parte de ese “leer” datos), en el sentido de que si las siguientes operaciones son para la

misma transacción (es decir, incluyen el mismo Id de Transacción) entonces estas operaciones serían llevadas a cabo en los datos leídos en este momento. No obstante, cualquier otro mensaje que no tenga esta transacción (es decir, que no incluye el mismo Id de Transacción) no sería capaz de acceder a estos datos "bloqueados". Estos datos "bloqueados" se muestran en la FIGURA 7 rodeados por una caja en negrita que indica que estos datos no pueden ser modificados/accedidos hasta que esta transacción particular que fue iniciada por el cliente 1 ha finalizado.

2b. El cliente 1 recibe una respuesta a leer (Entrada de Resultado de Búsqueda de LDAP/Búsqueda Completada) que incluye los datos 702 solicitados del directorio de LDAP 412. En este momento, el cliente 1 puede tomarse un tiempo para ejecutar alguna lógica interna, con cualquier propósito, como por ejemplo llevar a cabo alguna comprobación de consistencia en los datos leídos, conectarse a otro nodo para solicitar algún otro dato basándose en algo leído… En este caso, el cliente 1 ha actualizado los datos 702'.

3a. El cliente 2 envía una solicitud (Solicitud de Búsqueda de LDAP) de leer al menos una porción de los mismos datos 702 de la misma entrada al directorio de LDAP 412. Esta solicitud de leer podría ser una transacción que es diferente de la previa utilizada por el cliente 1(este tipo de solicitud no se muestra), o podría ser un mensaje independiente (como se muestra); independientemente de qué solicitud se utilice aplica todavía el mismo comportamiento.

3b. El cliente 2 recibe una respuesta a leer (Entrada de Resultado de Búsqueda de LDAP/Búsqueda Completada) que incluye los datos 702 solicitados. Esto es posible puesto que el mecanismo de solicitud de leer no es un mensaje de actualización. En este ejemplo, se asume que los mismos (o parte de) datos 702 "bloqueados" son leídos. A continuación, el cliente 2 ejecuta cualquier procesamiento y lógica requeridos, utilizando o no los datos leídos para tal propósito. En este caso, el cliente 2 ha actualizado los datos 702''.

4a. El cliente 2 envía una solicitud de actualización (Solicitud de Modificar de LDAP) solicitando la modificación de los datos "bloqueados" mediante la transacción del cliente 1. El directorio de LDAP 412 identifica que esta base de datos ha sido "bloqueada" como parte de una transacción previa, lo que significa que esta modificación particular no será permitida.

4b. El cliente 2 recibe un mensaje de error (Respuesta a Modificar de LDAP (fallida) indicando que la solicitud de modificación fue rechazada. En este ejemplo, la solicitud de modificación del cliente 2 fue rechazada con un mensaje de error. No obstante, el comportamiento específico cuando una modificación no está permitida puede variar; otra opción puede ser poner en cola la solicitud hasta que la transacción iniciada previa haya terminado. De nuevo, un propósito principal de este esquema es no permitir que el cliente 2 sobrescriba ningún dato 702 que haya sido leído por el cliente 1 y sea parte de una transacción en curso.

5a. El cliente 1 envía una solicitud de actualización (Solicitud de Modificar de LDAP) solicitando la modificación de los datos 702 como parte de la transacción en curso. En este ejemplo, debe asumirse que el cliente 1 envía una solicitud para modificar uno de varios atributos de los datos 702 "bloqueados", como parte de la transacción en curso, es decir, el mensaje de modificación incluye el correspondiente Id de Transacción (TransId) y los datos modificados 702'. El directorio de LDAP 412 identifica que esta solicitud está dentro de una transacción en curso y que por ello esta modificación está permitida en los datos 702 que fueron previamente "bloqueados" como parte de la misma transacción.

5b. El cliente 1 recibe un mensaje de éxito de modificación (Respuesta a Modificar de LDAP [Resultado Éxito]) del directorio de LDAP 412. Desde entonces, la modificación fue considerada parte de la misma transacción, por esa razón, fue procesada con éxito en la etapa 5a.

6a. El cliente 2 envía una solicitud (Solicitud de Detener Transacción (TransId1)) al directorio de LDAP 412 que cierra la transacción en curso. Esta solicitud puede ser un nuevo mensaje de LDAP que incluye el Id de transacción de la correspondiente transacción de LDAP que va a ser cerrada.

6b. El cliente 1 recibe un mensaje (Solicitud de Detener Transacción (TransId1)) indicando que el directorio de LDAP 412 cerró con éxito la transacción de LDAP. Este mensaje puede ser un nuevo mensaje de LDAP que confirma la ejecución con éxito del cierre de la solicitud de transacción. En realidad, esta etapa podría ser opcional, puesto que esta confirmación puede no ser necesaria en el lado del cliente.

Como puede verse, en este procedimiento se implementa un mecanismo de detección de colisión que evita que el cliente 2 sobrescriba algunos datos que el cliente 1 no desea que se modifiquen en este momento particular. En este esquema, los nuevos mensajes de LDAP han sido enviados en las etapas 1a, 1b, 6a y 6b así como un nuevo campo de control que incluye un Id de Transacción tal como se ha explicado anteriormente en las etapas 2a y 2b. Para más detalles acerca de la capacidad de transacción de LDAP que se utiliza para este mecanismo particular, se hace

referencia a una publicación de K. Zeilenga titulada “RFC 4528: Lightweight Directory Access Protocol (LDAP Assertion Control” de fecha de Junio de 2006.

En referencia a la FIGURA 8, hay un diagrama de secuencia de señal utilizado para ayudar a explicar otro mecanismo de detección de colisión de ejemplo que puede ser utilizado para detectar una colisión. Este mecanismo de detección de colisión hace uso de la capacidad de transacción de LDAP y del parámetro CDC mencionado anteriormente (que se define en cada una de las entradas) para validar la modificación de una de las múltiples entradas incluidas en la transacción como se describe a continuación:

1a & 2a. El cliente 1 envía dos solicitudes (Solicitudes de Búsqueda de LDAP) para leer algunos datos 802 y 804 y la información de CDC de las entradas 1 y 2 en el directorio de LDAP 412. Cada uno de estos mensajes de BÚSQUEDA de LDAP estándar podría solicitar leer cualquier cantidad de datos y el valor de CDC de las entradas 1 y 2.

1b & 2b. El cliente 1 recibe mensajes (dos Entrada de Resultado de Búsqueda de LDAP/Búsqueda Completada) del directorio de LDAP 412 incluyendo los datos 802 y 804 solicitados en las entradas 1 y 2 y sus correspondientes valores de CDC. En este momento, el cliente 1 se toma un tiempo para ejecutar alguna lógica interna, con cualquier propósito, como por ejemplo, llevar a cabo algunas comprobaciones de consistencia sobre los datos leídos, conectarse a otro nodo para solicitar algún otro dato basándose en algo leído… En este caso, el cliente 1 ha actualizado los datos 802’ y 804’.

3a. El cliente 2 envía una solicitud (Solicitud de Búsqueda de LDAP) para leer algunos datos 804 de la entrada 2 en el directorio de LDAP 412. Esta solicitud podría ser una transacción que sea diferente de la previa utilizada por el cliente 2 (este tipo de solicitud no se muestra), o podría ser un mensaje independiente (no se muestra); independientemente de qué solicitud se utilice aplica todavía el mismo comportamiento. En este ejemplo, se asume que los mismos (o parte de) los datos “bloqueados” 804 son leídos.

3b. El cliente 2 recibe un mensaje (Entrada de Resultado de Búsqueda de LDAP/Búsqueda Completada) del directorio de LDAP 412 que incluye datos solicitados en la entrada 2 y el correspondiente valor de CDC. A continuación, el cliente 2 ejecuta cualquier procesamiento y lógica, utilizando o no los datos leídos para tal propósito. En este caso, el cliente 2 ha actualizado los datos 804’’.

4a. El cliente 2 envía un mensaje (Solicitud de Modificar de LDAP) que requiere la modificación de los datos en la entrada 1 del directorio de LDAP 412. Se solicita que varios atributos en la entrada 2 sean modificados, incluyendo la modificación del valor de CDC para la entrada 2. La modificación del valor de CDC se realiza utilizando el mismo proceso descrito anteriormente con respecto a la primera realización de la presente invención. No se abre ninguna transacción en este caso, porque sólo se impacta una entrada.

4b. El cliente 2 recibe un mensaje (Respuesta a Modificar de LDAP [Resultado Éxito]) desde el directorio de LDAP 412, indicando que la solicitud de modificación fue aceptada. La modificación solicitada en la etapa 4a fue aceptada, y los datos modificados 804’’ fueron almacenados en la entrada 2 y el valor de CDC para la entrada 2 fue actualizado.

5a. El cliente 1 envía un mensaje (Iniciar Solicitud de Transacción) solicitando el inicio de una transacción con el directorio de LDAP 412. Este mensaje también solicita que el directorio de LDAP 412 genere un identificador de transacción.

5b. El cliente 1 recibe un mensaje (Iniciar Respuesta a Transacción (TransId1)) en respuesta a la solicitud de inicio de transacción en el directorio de LDAP 412. En particular, el directorio de LDAP 412 devuelve un identificador de transacción (TransId1).

6a. El cliente 1 envía un mensaje (Solicitud de Modificar de LDAP (TransId1)) para solicitar una modificación de datos en la entrada 1 del directorio de LDAP 412. Se solicita la modificación de varios atributos en la entrada 1, incluyendo la modificación del valor de CDC para la entrada 1. La modificación del valor de CDC se lleva a cabo utilizando el mismo proceso descrito anteriormente con respecto a la primera realización de la presente invención.

6b. El directorio de LDAP 412 acepta la solicitud de modificación del cliente 1 puesto que el valor de CDC no ha sido cambiado para la correspondiente entrada 1. Pero, la operación de modificación sigue pendiente de ser ejecutada cuando la transacción completa sea aceptada.

7a. El cliente 1 envía un mensaje (Solicitud de Modificar de LDAP (TransId1)) para solicitar una modificación de los datos en la entrada 2 del directorio de LDAP 412. Se solicita la modificación de varios atributos de la entrada 2, incluyendo la modificación del valor de CDC para la entrada 2. La modificación del valor de CDC se

realiza utilizando el mismo proceso descrito anteriormente con respecto a la primera realización de la presente invención.

7b. El cliente 1 recibe un mensaje (Respuesta a Modificar de LDAP [Resultado Fallido]) indicando que la solicitud de modificación para la entrada 2 fue rechazada por el directorio de LDAP 412. El directorio de LDAP 412 rechazó esta modificación porque en este ejemplo el valor de CDC para la entrada 2 fue modificado por el proceso ejecutado por el cliente 2 durante la etapa 4a.

8a. El cliente 1 envía un mensaje (Solicitud de Detener Transacción [Abortar TransId1]) para 'abortar' la transacción en curso porque una parte de ella no puede ser ejecutada. Éste puede ser un nuevo mensaje de LDAP que incluye el Id de Transacción de la correspondiente transacción que va a ser cerrada.

8b. El cliente 1 recibe un mensaje (Detener Respuesta a Transacción [TransId1]) desde el directorio de LDAP 412 indicando que las transacciones han sido cerradas correctamente. Éste puede ser un nuevo mensaje de LDAP para confirmar la ejecución con éxito de la solicitud de transacción que se cierra. Alternativamente, éste podría ser un mensaje opcional puesto que la confirmación puede no ser necesaria en el lado del cliente.

En referencia a la FIGURA 9, hay un diagrama de secuencia de señal utilizado para ayudar a explicar otro mecanismo de detección de colisión de ejemplo que puede ser utilizado para detectar una colisión. Este mecanismo de detección de colisión hace uso de Aserciones de LDAP que permiten a los clientes y en particular a los clientes LDAPv3, especificar una condición que debe ser verdadera para que la operación sea ejecutada por el directorio de LDAP 412 o si no que la operación solicitada sea rechazada como un conjunto. Las Aserciones de LDAP han sido definidas en el documento RFC 4528 mencionado anteriormente y pueden encontrarse detalles acerca de clientes de LDAPv3 en el siguiente documento: K. Zeilenga "RFC 4510: Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map", de Junio de 2006.

En particular, las Aserciones de LDAP se utilizan para comprobar una condición en la misma operación en la que la modificación se lleva a cabo, y esto resulta en una atomicidad de la operación puesto que tanto la condición de validación como la de modificación de datos son llevadas a cabo sin la interrupción de ninguna otra operación. Para permitir esto, el RFC 4528 definió un nuevo control de LDAP (el "control de afirmación") y el documento de K. Zeilenga "RFC 4521: Considerations for Lightweight Directory Access Protocol (LDAP) Extensions" de Junio de 2006 explica cómo puede este mecanismo basado en control ser extendido a las operaciones de LDAPv3. Por ejemplo, este control puede ser incorporado a cualquier operación de actualización de LDAPv3 (es decir, Añadir, Borrar, Modificar y Modificar DN) para soportar la adición, el borrado, la modificación y el renombrado condicionales del objeto de objetivo. Esta condición afirmada es también evaluada como una parte integral de la operación de LDAPv3 implicada. El RFC 4528 también especifica que la entrada de objetivo en el DIT, es decir, a la que debe aplicarse la comprobación de la afirmación (cuando se recibe), es siempre una única entrada en el DIT, que es el campo de entrada o de objeto en la solicitud de LDAP (como simplificación general).

Esta realización particular del mecanismo de detección de colisión utiliza Imposiciones de LDAP y también define y añade un nuevo atributo entero de un solo valor del tipo de "Número de Secuencia" para cada entrada que requiere detección de colisión. Este nuevo atributo de tipo entero de un solo valor "Número de Secuencia" puede ser gestionado como un contador cíclico. Este nuevo atributo entero con un solo valor puede también ser declarado como un atributo "mandatorio" en las clases de objeto "de estructura" deseadas (es decir, las clases de objeto utilizadas para entradas a directorio) o puede ser definido en una nueva clase de objeto "auxiliar". Para implementar este mecanismo de detección de colisión y resolver el problema de la modificación, cada cliente de LDAP 402, 404, 408 y 410 llevaría a cabo las siguientes etapas:

1. Leer una o varias entradas, incluyendo al menos una entrada que es una preparada para detección de colisión (es decir, incluye el tipo de atributo Número de Secuencia).

2. Aplicar la lógica comercial utilizando los datos obtenidos.

3. Solicitar que los datos sean actualizados en una entrada definida para la detección de colisión. La solicitud incluiría las siguientes operaciones de Solicitud de Modificar de LDAP:

3.1 Una "condición de afirmación" que comprueba que el atributo de tipo Número de Secuencia no tiene el valor que fue leído en la etapa 1.

3.2 Una operación de "reemplazar" para incrementar el valor de detección de colisión almacenado en el atributo de tipo Número de Secuencia.

NOTA: La afirmación se comprueba antes de que se lleve a cabo ninguna modificación solicitada (añadir/borrar/reemplazar). Si la comprobación de afirmación falla entonces toda la operación de Solicitud de Modificar de LDAP es descartada.

El diagrama de flujo de señal de la FIGURA 9 ilustra un escenario de ejemplo que utiliza este mecanismo de detección de colisión particular de la presente invención:

1a. El cliente 1 envía un mensaje (Solicitud de Búsqueda de LDAP) solicitando leer datos 902 de una o varias entradas, incluyendo al menos una entrada 1 que está preparada para detección de colisión. El mensaje también tiene una solicitud de leer el atributo de tipo Número de Secuencia dentro de al menos una entrada 1 del directorio de LDAP 412.

1b. El cliente recibe los datos solicitados 902 y el valor del atributo Número de Secuencia desde el directorio de LDAP 412. Esto puede ser llevado a cabo por medio de una Entrada de Resultado de Búsqueda de LDAP que incluye la información solicitada y el Número de Secuencia y por medio de un mensaje de Resultado de Búsqueda de LDAP Completada que indica que la búsqueda tuvo éxito. En este momento el cliente 1 se toma un tiempo para ejecutar alguna lógica interna, con cualquier propósito, como por ejemplo, llevar a cabo alguna comprobación de consistencia en los datos leídos, conectarse a otro nodo para solicitar algún dato más basándose en algo leído… En este caso, el cliente 1 ha actualizado los datos 902'.

2a. El cliente 2 envía un mensaje (Solicitud de Búsqueda de LDAP) solicitando leer los datos 902 de una o varias entradas, incluyendo al menos una entrada 1 que está preparada para la detección de colisión. El mensaje tiene también una solicitud de leer el tipo de atributo Número de Secuencia dentro de al menos una entrada 1 del directorio de LDAP 412. En este ejemplo, el cliente 1 y el cliente 2 están interesados en modificar los datos en la misma entrada (no es necesario que sea el mismo atributo, pero sólo la misma entrada, es decir, cualquiera de los atributos para cualquiera de las Clases de Objeto para esa entrada particular).

2b. El cliente 2 recibe los datos solicitados 902 y el valor del atributo Número de Secuencia del directorio de LDAP 412. Esto puede ser realizado por medio de un mensaje de Entrada de Resultado de Búsqueda de LDAP que incluye la información solicitada y el Número de Secuencia y por medio de un mensaje de Resultado de Búsqueda de LDAP Completada que indica que la búsqueda tuvo éxito. En este momento, el cliente 2 se toma un tiempo para ejecutar alguna lógica interna, con cualquier propósito, como por ejemplo llevar a cabo alguna comprobación de consistencia en los datos leídos, conectarse a otro nodo para solicitar algún dato más basándose en algo leído… En este caso, el cliente 2 ha actualizado los datos 902''.

3a. El cliente 2 envía un mensaje (Solicitud de Modificar de LDAP) al directorio de LDAP 412 solicitando una actualización de los datos en una entrada lista para la detección de colisión, donde el número de secuencia originalmente leído fue "n". En particular, el cliente 2 envía una Solicitud de Modificar de LDAP que incluye: (1) una operación de afirmación: Afirmación: Número de Secuencia = n, críticamente: VERDADERO; (2) datos modificados 902''; y (3) una operación de reemplazo: REEMPLAZAR: Número de Secuencia = n+1 (nota: la etapa 1 debe ser verdadera antes de que las etapas 2 y 3 puedan ser llevadas a cabo). Básicamente, el cliente 2 envía una Solicitud de Modificar de LDAP que incluye la actualización del atributo número de secuencia (incrementando su valor en 1), y al mismo tiempo, la Solicitud de Modificar de LDAP incluye como una "afirmación" para comprobar el número de secuencia para asegurar que es el mismo que se acababa de leer en la etapa 2b. Sólo si la evaluación de la afirmación es VERDADERA, entonces se lleva a cabo la operación de Modificar de LDAP. Y, esto sucedería sólo si el número de secuencia no ha sido incrementado por algún otro cliente, el cual si hubiese sido incrementado revelaría que el otro cliente ha modificado los datos desde que fueron leídos en la etapa 2b.

3b. El cliente 2 recibe una Respuesta a Modificar de LDAP desde el directorio de LDAP 412 indicando que la solicitud de modificación tuvo éxito.

4a. El cliente 1 envía un mensaje (Solicitud de Modificación de LDAP) al directorio de LDAP 412 solicitando una actualización de los datos en una entrada preparada para la detección de colisión, donde el número de secuencia originalmente leído fue "n". En particular, el cliente 1 envía una Solicitud de Modificar de LDAP que incluye: (1) una operación de afirmación: Afirmación: Número de Secuencia = n críticamente: VERDADERO; (2) datos modificados 902'; y (3) una operación de reemplazo: REEMPLAZAR: Número de Secuencia n+1 (nota: la etapa 1 debe ser verdadera antes de que las etapas 2 y 3 puedan ser llevadas a cabo). En este caso, el directorio de LDAP 412 determina que la afirmación es evaluada como FALSO, puesto que el atributo número de secuencia no tiene el mismo valor que este cliente originalmente leyó durante la etapa 1b. Esto significa que se ha evitado una colisión.

4b. El cliente 1 recibe una Respuesta a Modificar de LDAP desde el directorio de LDAP 412 indicando que la solicitud de actualización no tuvo éxito. Así, el cliente 1 sabe que otro cliente ha actualizado los datos durante el tiempo en que estaba procesando los datos leídos.

De lo anterior, resultará evidente que se han explicado cuatro mecanismos diferentes de detección de colisión que permiten que un primer cliente de LDAP lleve a cabo una Búsqueda de LDAP, procese la respuesta de LDAP y a continuación envíe modificaciones (Modificar de LDAP) al directorio de LDAP 412 estando seguro de que la modificación solicitada no será llevada a cabo si otro cliente de LDAP hubiese llevado a cabo previamente una modificación en los datos que fueron originalmente leídos por el primer cliente de LDAP. Esto resulta deseable puesto que de esta manera no hay creaciones de inconsistencias de datos. La presente invención tiene varias ventajas, algunas de las cuales se listan a continuación como sigue:

1. La detección de la modificación de otro cliente en algunos datos leídos anteriormente evita la creación de inconsistencias de datos. Esto es, si un cliente requiere la validación de algunos datos para llevar a cabo una actualización, si los datos son modificados, entonces esas actualizaciones pueden provocar un problema si esos datos son finalmente actualizados. En otras palabras, un cliente podría asegurarse de que las modificaciones que dependen de ciertos valores (situados en alguna parte del DIT) son llevadas a cabo de manera consistente (es decir, los datos relevantes no son modificados).

2. Cualquier número de clientes de LDAP concurrentes podría ser detectado.

3. Las colisiones son detectadas (y evitadas) sin la necesidad de ningún mecanismo de bloqueo/desbloqueo con efectos colaterales de “punto muerto” no deseables.

4. Los mecanismos de detección de colisión basados en extender/definir nuevos mensajes de LDAP (como es el caso para el basado en afirmación) tienen una elevada dependencia en el lado del cliente de LDAP de la tecnología del directorio de LDAP final que se está utilizando en el sistema.

5. Los mecanismos de detección de colisión de la presente invención son también válidos para los sistemas de directorio X.500, puesto que proporcionan una Puerta de Enlace de Acceso de LDAP que permite la comunicación con los clientes de LDAP estándar.

7. Los mecanismos de detección de colisión no están limitados a su uso en el campo de la comunicación, sino que también pueden ser aplicados a cualquier aplicación en cualquier campo que implique el uso de clientes de LDAP y un directorio de LDAP.

8. Los mecanismos de detección de colisión tienen las siguientes ventajas:

- Un cliente de LDAP estándar es capaz de detectar si ocurre una colisión (es decir, los datos leídos por este cliente pueden haber sido actualizados por otro cliente).
- Los mecanismos de CDC de la presente invención se basan en parte en el LDAP estándar que es soportado hoy en día por cualquier implementación de servidor de LDAP comercial. No requiere ningún soporte de LDAP ampliado en el directorio de LDAP o en los clientes de LDAP.
- Los mecanismos de CDC de la presente invención implementan un mecanismo exclusivo mutuo (mutex), donde el recurso común es los datos (para ser leídos y modificados) y la sección crítica es el tiempo desde el momento en que los datos son leídos por un cliente de LDAP y el momento en que el mismo cliente de LDAP requiere modificaciones de datos.

9. El mecanismo de detección de colisión de Afirmación de LDAP tiene las siguientes ventajas:

- Un cliente de LDPA estándar es capaz de detectar si ocurre una colisión (es decir, los datos leídos por este cliente pueden haber sido actualizados por otro cliente).
- El mecanismo de Afirmación de LDAP de la presente invención se basa en parte en el LDAP estándar. No requiere ningún soporte de LDAP ampliado en ningún servidor de LDAP o en el cliente de LDAP.

Para una explicación más detallada acerca de los principios básicos de la tecnología de LDAP, se hace referencia a los siguientes documentos:

1. K. Zeilenga “RFC 4510: Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map”, Junio de 2006.

2. J. Sermersheim “RFC 4511: Lightweight Directory Access Protocol (LDAP): The Protocol”, Junio de 2006.

3. K. Zeilenga “RFC 4521: Considerations for Lightweight Directory Access Protocol (LDAP) Extensions”, Junio de 2006.

4. K. Zeilenga “RFC 4528: Lightweight Directory Access Protocol (LDAP) Assertion Control”, Junio de 2006.

5. Lista de Correos Electrónicos del Software OpenLDAP: Mensaje00529 (definir/uso de tipos de atributo “bloqueo”) (descargado de http://www.openldap.org/lists/openldap-software/200301/msg00529.html).

6. K. Zeilenga “LDAP Transactions”, 8 de Julio de 2007 (descargado de http://www.ietf.org/internet-drafts/draft-zeilenga-ldap-txn-10.txt).

En resumen, el directorio 412 que es accesible para un número de clientes 402, 404, 406, 408 y 410 para leer y actualizar datos en él está también dispuesto para detectar y evitar una colisión de operación de actualización en una entrada en él implementando los esquemas mostrados en las FIGURAS 5 y 8-9 y utilizando los siguientes componentes (véase la FIGURA 4): un medio de lectura de entrada 418 que recibe una solicitud del cliente para leer datos en una entrada al directorio; un medio de procesamiento 420 que asigna al menos un valor de detección de colisión dado correspondiente al menos a un subconjunto de los datos en la entrada al directorio; un medio de lectura de salida 422 que presenta hacia el cliente los datos solicitados junto con el al menos un valor de detección de colisión dado; un medio de actualización de entrada 424 que recibe una solicitud del cliente de modificar el al menos un subconjunto de los datos en la entrada al directorio, donde la solicitud incluye al menos un valor de detección de colisión actualizado para cada al menos un valor de detección de colisión dado; un detector de colisión 426 que determina si el al menos un valor de detección de colisión actualizado está de acuerdo con un correspondiente valor de detección de colisión actual o no; un medio de actualización de salida 428 que acepta y ejecuta la solicitud de modificar el al menos un subconjunto de los datos en la entrada al directorio, donde el al menos un valor de de detección de colisión actualizado está de acuerdo con el correspondiente valor de detección de colisión actual, o denegando y rechazando si no la solicitud; y asignando el medio de procesamiento 420 al valor de detección de colisión actual un siguiente valor, donde la solicitud se encuentra aceptable.

Además, el ordenador frontal 408 del cliente (por ejemplo) puede acceder a un directorio 412 para leer y actualizar datos en él de acuerdo con los esquemas mostrados en las FIGURAS 5 y 8-9, utilizando los siguientes componentes (véase la FIGURA 4): un medio de lectura de salida 430 para enviar una solicitud de leer datos en una entrada en el directorio; un medio de lectura de entrada 432 para obtener los datos solicitados junto con al menos un valor de detección de colisión dado correspondiente al menos un subconjunto de los datos en la entrada del directorio; un procesador 434 para procesar los datos obtenidos; un medio de actualización de salida 436 para enviar una solicitud de modificar el al menos un subconjunto de los datos en la entrada al directorio, donde la solicitud incluye al menos un valor de detección de colisión actualizado para cada al menos un valor de detección de colisión dado; y un medio de actualización de entrada 438 para recibir una aceptación de la solicitud de modificar el al menos un subconjunto de los datos en la entrada del directorio, donde el al menos un valor de detección de colisión actualizado está de acuerdo con un correspondiente valor de detección de colisión actual en el directorio, o si no una denegación de la solicitud.

Aunque se han ilustrado varias realizaciones de la presente invención en los Dibujos que se acompañan y se han descrito en la Descripción Detallada anterior, debe entenderse que la invención no está limitada a la realización explicada, sino que por el contrario también es capaz de numerosas redisposiciones, modificaciones y sustituciones sin separarse del alcance de la invención, tal como se explica y define mediante las reivindicaciones siguientes.

REIVINDICACIONES

1. Un método para detectar y evitar colisiones en una entrada en un directorio con Protocolo Ligero de Acceso a Directorios (LDAP" (Lightweight Directory Access Protocol, en inglés), mediante operaciones de actualización desde el ordenador frontal de más de un cliente (FE DE HSS, FE DE HLR), comprendiendo el citado método las etapas de:

- recibir en el directorio una solicitud del ordenador frontal de un cliente de leer datos en una entrada al directorio;
- asignar en el directorio al menos un valor de detección de colisión dado para al menos un subconjunto de los datos en la entrada al directorio, donde al al menos un valor de detección de colisión dado se le proporciona un atributo de contador de detección de colisión, CDC (Collision Detection Counter, en inglés), que tiene un valor (x) en la entrada al directorio;
- enviar hacia el ordenador frontal del cliente los datos solicitados junto con el al menos un valor de detección de colisión dado;
- recibir en el directorio una solicitud desde el ordenador frontal de un cliente de modificar el al menos un subconjunto de los datos en la entrada al directorio, teniendo la solicitud al menos un valor de detección de colisión actualizado para cada al menos un valor de detección de colisión dado, donde la solicitud que tiene el al menos un valor de detección de colisión actualizado incluye:

(i) una o más operaciones de AÑADIR como sigue AÑADIR: CDC = (x) + 1, AÑADIR: CDC = (x) + 2… AÑADIR: CDC = (x) + Número de Colisiones para Detectar, NCD, donde NCD tiene un valor igual o mayor que el número de clientes concurrentes que podrían posiblemente leer y modificar los datos de la entrada al directorio;
(ii) datos modificados; y
(iii) una operación de REEMPLAZAR como sigue: REEMPLAZAR: CDC = (x) + 1; determinando en el directorio si la solicitud incluye una operación de AÑADIR que intente operar en un atributo de CDC actual asignado al al menos un subconjunto de datos y que tiene un valor diferente dentro de la entrada al al menos un valor de atributo de CDC (x) dado enviado al ordenador frontal del cliente;

- si el resultado de la etapa de determinación es negativo, enviar hacia el ordenador frontal del cliente una aceptación de la solicitud de modificar el al menos un subconjunto de los datos en la entrada al directorio y asignar al valor de detección de colisión actual en la entrada un valor siguiente (x)+1, y si no enviar la denegación de la solicitud hacia el ordenador frontal del cliente.

2. El método de la reivindicación 1, en el que el ordenador frontal del cliente inicia una transacción con el directorio antes de enviar la solicitud de modificar los datos.

3. El método de la reivindicación 1, en el que el atributo de CDC es un atributo mandatorio en una clase de objetos de estructura deseados o en una nueva clase de objeto.

4. El método de la reivindicación 1, en el que el atributo de CDC es un contador cíclico.

5. Un directorio al que acceden un número de clientes con un Protocolo Ligero de Acceso a Directorios "LDAP" (Lightweight Directory Access Protocol, en inglés) para leer y actualizar los datos en él, y dispuesto para detectar y evitar una colisión de operación de actualización en una entrada al directorio, comprendiendo el directorio:

un medio de lectura de entrada que recibe una solicitud desde un ordenador frontal del cliente para leer datos en una entrada al directorio;
un medio de procesamiento que asigna al menos un valor de detección de colisión dado a al menos un subconjunto de los datos en la entrada al directorio, donde el al menos un valor de detección de colisión dado es un atributo de contador de detección de colisión dado, CDC, que tiene un valor (x) en la entrada al directorio;
un medio de lectura de salida que presenta hacia el ordenador frontal del cliente los datos solicitados junto con el al menos un valor de detección de colisión dado;
un medio de actualización de entrada que recibe una solicitud desde el ordenador frontal del cliente para modificar el al menos un subconjunto de los datos en la entrada al directorio, teniendo la solicitud al menos un valor de detección de colisión actualizado para cada al menos un valor de detección de colisión dado, donde la solicitud que tiene el al menos un valor de detección de colisión actualizado incluye:

(i) una o más operaciones de AÑADIR como sigue AÑADIR: CDC = (x) + 1, AÑADIR: CDC = (x) + 2… AÑADIR: CDC = (x) + Número de Colisiones para Detectar, NCD, donde NCD tiene un valor igual o mayor que un número de clientes concurrentes que podrían posiblemente leer y modificar los datos desde la entrada al directorio;
(ii) datos modificados; y

(iii) una operación de REEMPLAZAR como sigue: REEMPLAZAR CDC = (x) + 1;

determinando un detector de colisión si la solicitud incluye una operación de AÑADIR que intenta operar sobre un atributo de CDC actual asignado al menos a un subconjunto de datos y que tiene un valor diferente dentro de la entrada al al menos un valor (x) de atributo de CDC dado enviado al ordenador frontal del cliente; si el resultado de la determinación mediante el detector de colisión es negativo, aceptando y ejecutando un medio de actualización de salida la solicitud de modificar el al menos un subconjunto de los datos en la entrada al directorio, y asignando un medio de procesamiento al valor de detección de colisión actual un valor siguiente; y si no, denegando y rechazando el medio de actualización de salida la solicitud hacia el ordenador frontal del cliente.

6. El directorio de la reivindicación 5, en el que el medio de lectura de entrada está adaptado para recibir desde el ordenador frontal del cliente una solicitud de iniciar una transacción antes de enviar la solicitud de modificar los datos.

7. El directorio de la reivindicación 5, en el que el atributo de CDC es un atributo mandatorio en una clase de objeto de estructura deseada o en una nueva clase de objeto auxiliar.

8. El directorio de la reivindicación 5, en el que el atributo de CDC es un contador cíclico.

9. Un ordenador frontal (FE DE HSS, FE DE HLR) de cliente para acceder a un directorio con un Protocolo Ligero de Acceso a Directorios “LDAP” (Lightweight Directory Access Protocol, en inglés) para leer y actualizar los datos en él, y que comprende:

un medio de lectura de salida para enviar una solicitud de leer datos en una entrada al directorio;
un medio de lectura de entrada para recibir los datos solicitados junto con al menos un valor de detección de colisión dado para al menos un subconjunto de los datos en la entrada al directorio, donde el al menos un valor de detección de colisión dado es un atributo de contador de detección de colisión dado, CDC, que tiene un valor (x) en la entrada al directorio;
un procesador para procesar los datos recibidos;
un medio de actualización de salida para enviar una solicitud de modificar el al menos un subconjunto de los datos en la entrada al directorio, teniendo la solicitud al menos un valor de detección de colisión actualizado para cada al menos un valor de detección de colisión dado, donde la solicitud que tiene el al menos un valor de detección de colisión actualizado incluye:

(i) una o más operaciones de AÑADIR: CDC = (x) + 1, AÄDIR: CDC = (x) + 2… AÑADIR: CDC = (x) + Número de Colisiones para Detectar, NCD, donde NCD tiene un valor igual o mayor que un número de clientes concurrentes que podrían posiblemente leer y modificar los datos desde la entrada al directorio;
(ii) datos modificados; y
(iii) una operación de REEMPLAZAR como sigue: REEMPLAZAR CDC = (x) + 1; y

un medio de actualización de entrada para recibir la denegación de la solicitud, si la solicitud incluye una operación de AÑADIR que intenta operar en un atributo de CDC actual asignado al al menos un subconjunto de datos que tienen un valor diferente dentro de la entrada que el al menos un valor de atributo de CDC (x) dado recibido en el medio de lectura de entrada; y para recibir, si no, la aceptación de la solicitud para modificar el al menos un subconjunto de los datos en la entrada al directorio.

10. El ordenador frontal de cliente de la reivindicación 9, en el que el medio de lectura de salida está también adaptado para enviar hacia el directorio una solicitud de inicio de una transacción antes de enviar la solicitud de modificar los datos.

11. El ordenador frontal de cliente de la reivindicación 9, en el que el atributo de CDC es un atributo mandatorio en una clase de objeto de estructura o en una nueva clase de objeto auxiliar.

12. El ordenador frontal de cliente de la reivindicación 9, en el que el atributo de CDC es un contador cíclico.

13. Un método para detectar y evitar colisiones en una entrada a un directorio a la que se accede con un Protocolo Ligero de Acceso a Directorios “LDAP” (Lightweight Directory Access Protocol, en inglés), mediante una operación de actualización desde el ordenador frontal (FE DE HSS, FE DE HLR) de un ordenador frontal de un cliente, comprendiendo el citado método las etapas de:

enviar una solicitud desde el ordenador frontal de un cliente para leer datos en una entrada al directorio;

recibir en el ordenador frontal del cliente los datos solicitados junto con al menos un valor de detección de colisión dado para al menos un subconjunto de los datos, en el que el al menos un valor de detección de colisión dado es un atributo de contador de detección de colisión, CDC, que tiene un valor (x) en la entrada al directorio;

procesar los datos obtenidos en el ordenador frontal del cliente;

enviar una solicitud desde el ordenador frontal del cliente para modificar el al menos un subconjunto de los datos en la entrada al directorio, teniendo la solicitud al menos un valor de detección de colisión actualizado para cada al menos un valor de detección de colisión dado, donde la solicitud que tiene el al menos un valor de detección de colisión incluye:

(i) una o más operaciones de AÑADIR como sigue: AÑADIR: CDC = (x) + 1, AÑADIR: CDC = (x) + 2… AÑADIR: CDC = (x) + Número de Colisiones para Detectar, NCD, donde NCD tiene un valor igual o mayor que el número de clientes concurrentes que podrían posiblemente leer y modificar los datos desde la entrada al directorio;

(ii) datos modificados; y

(iii) una operación de REEMPLAZAR como sigue: REEMPLAZAR CDC = (x) + 1; y

recibir en el ordenador frontal del cliente la denegación de la solicitud, si la solicitud incluye una operación de AÑADIR que intenta operar en un atributo de CDC actual asignado al al menos un subconjunto de datos y que tiene un valor diferente dentro de la entrada del al menos un valor de atributo de CDC (x) dado recibido en el ordenador frontal del cliente, y para recibir, si no, la aceptación de la solicitud de modificar el al menos un subconjunto de los datos en la entrada al directorio.

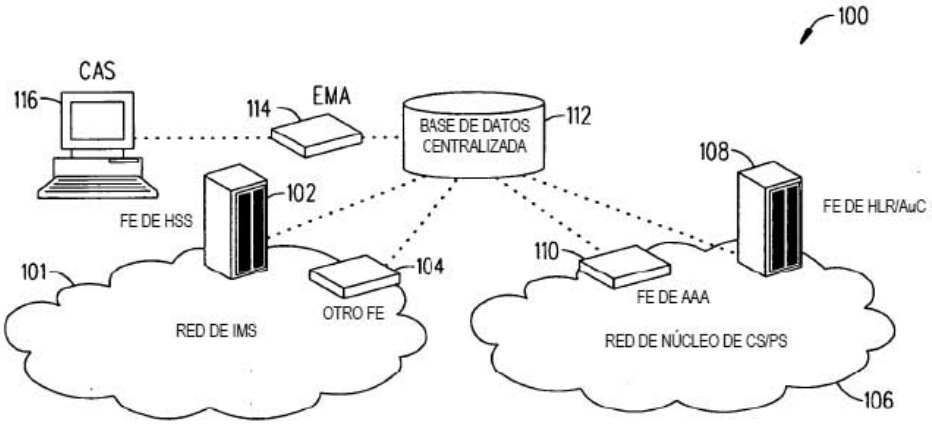


FIG. 1 (TÉCNICA ANTERIOR)

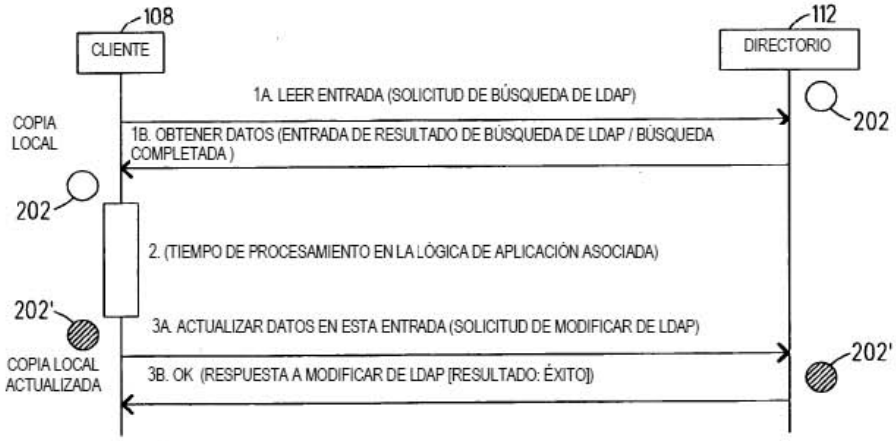


FIG. 2 (TÉCNICA ANTERIOR)

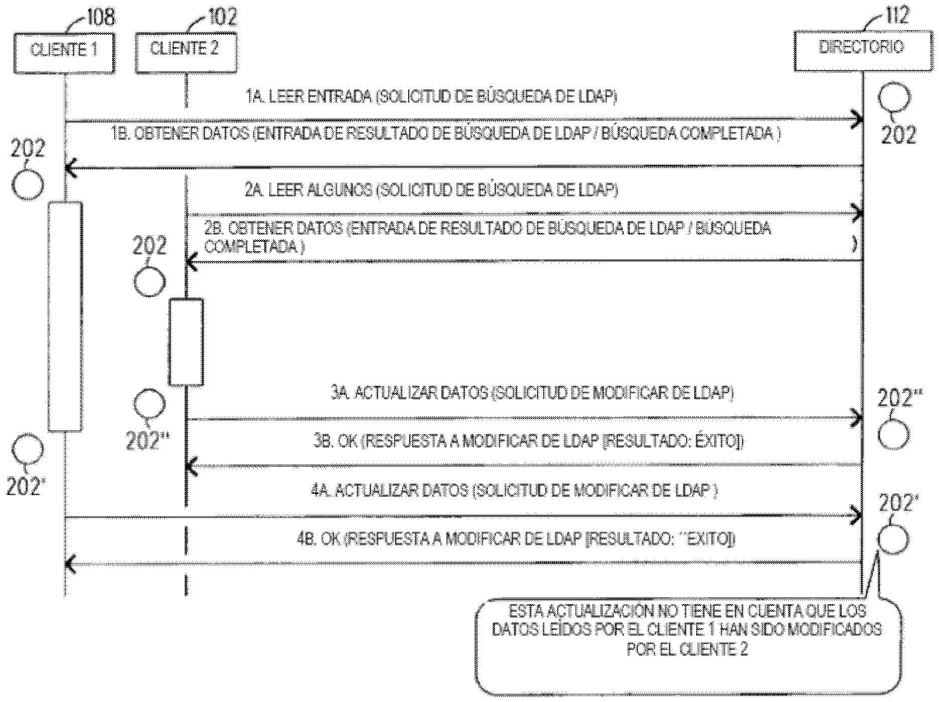


FIG. 3 (TÉCNICA ANTERIOR)

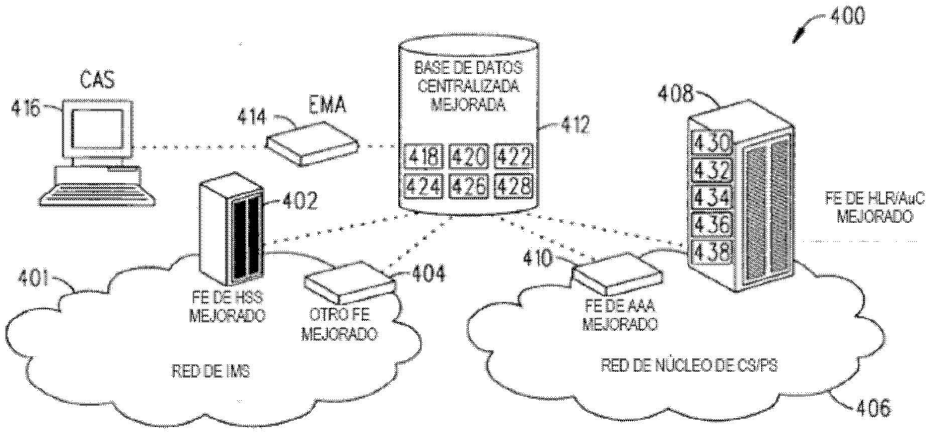
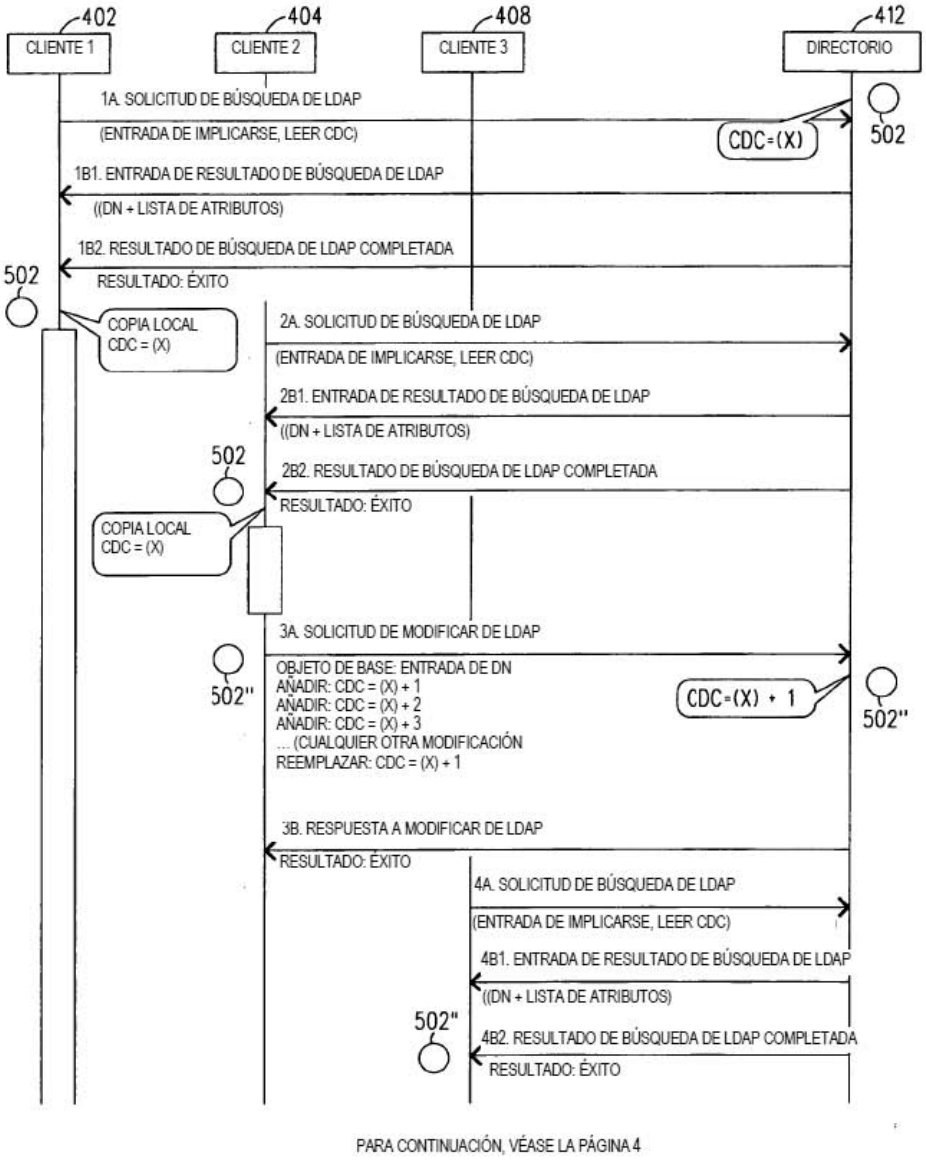


FIG. 4



PARA CONTINUACIÓN, VÉASE LA PÁGINA 4

FIG. 5

PARA CONTINUACIÓN, VÉASE LA PÁGINA 3

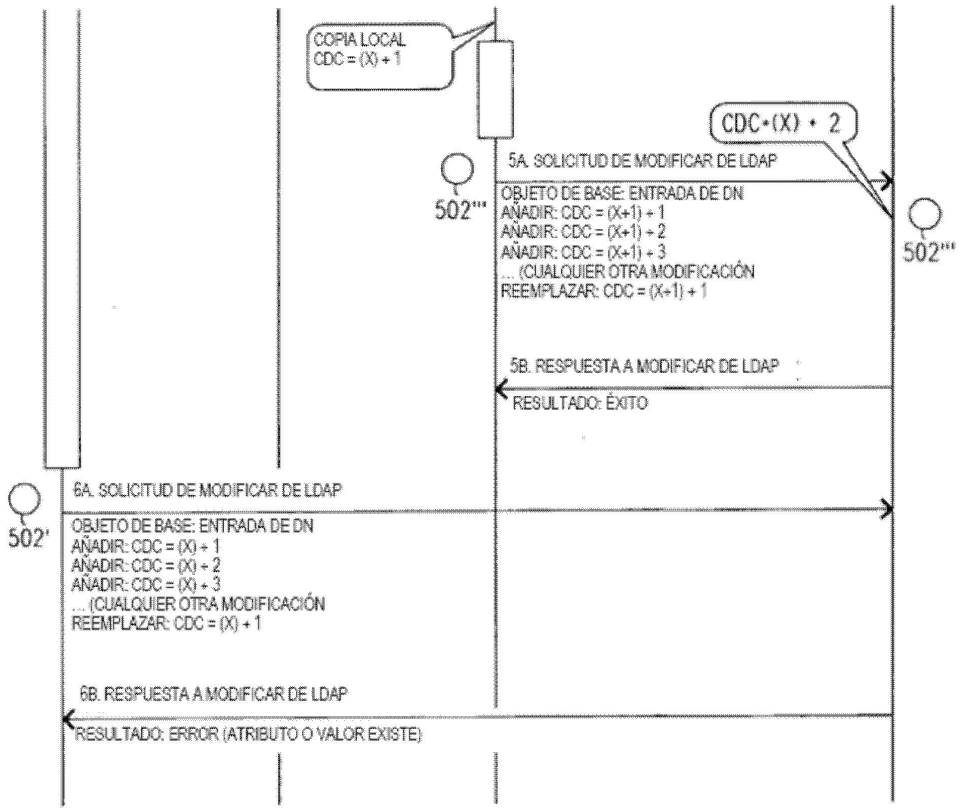
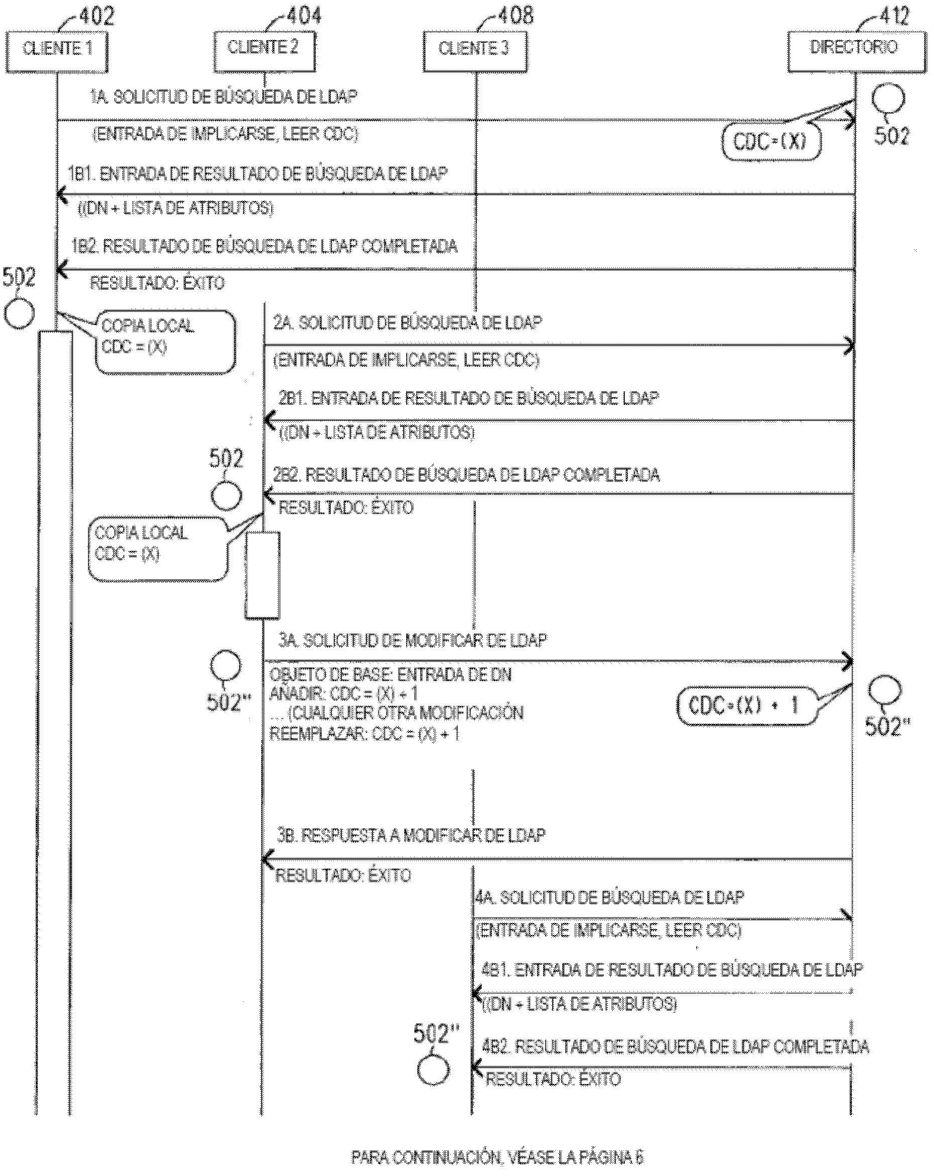


FIG. 5 CONTINUACIÓN



PARA CONTINUACIÓN, VÉASE LA PÁGINA 6

FIG. 6

PARA CONTINUACIÓN, VÉASE LA PÁGINA 5

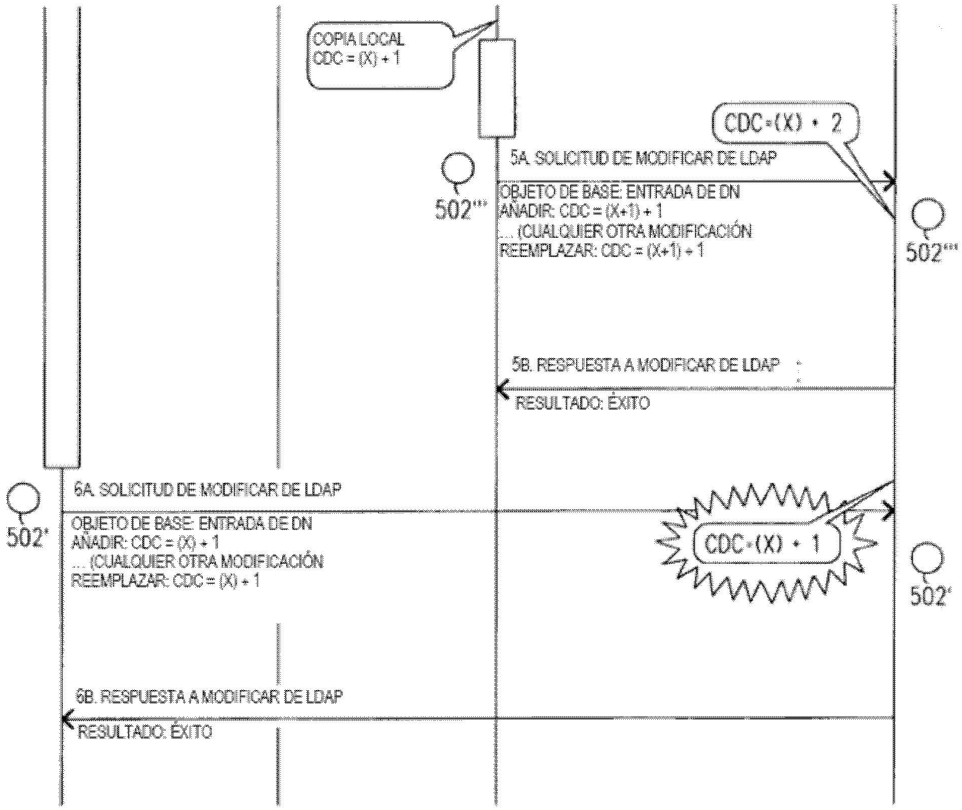


FIG. 6 CONTINUACIÓN

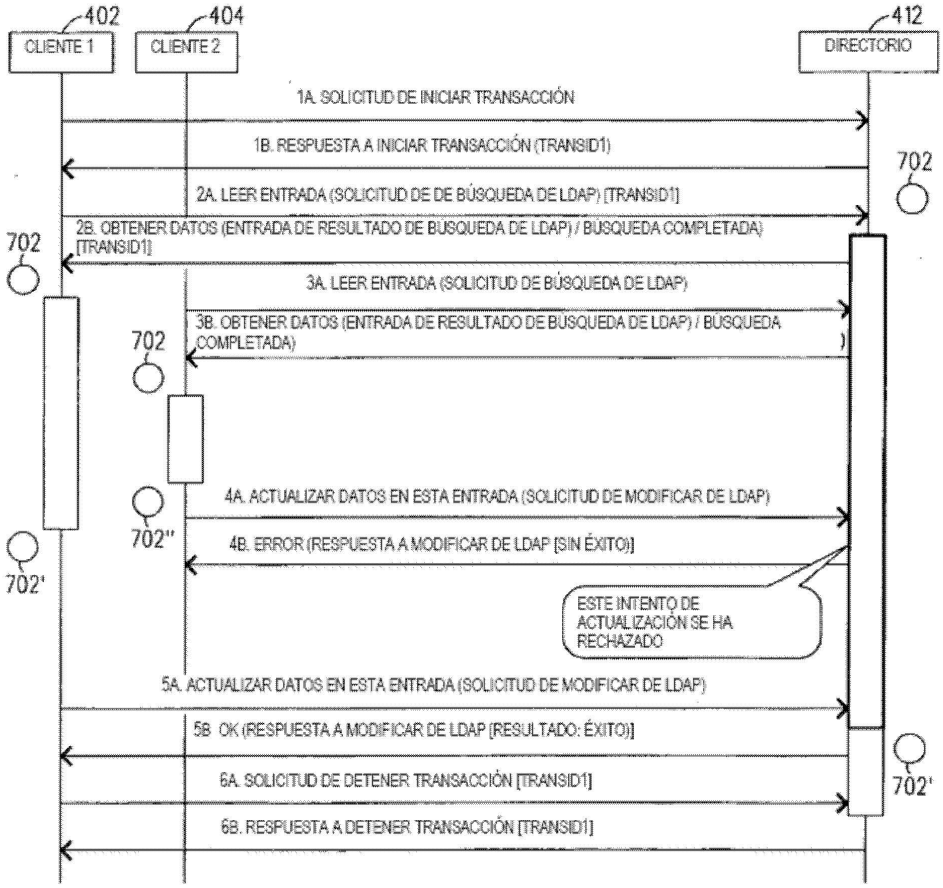

402
CLIENTE 1
404
CLIENTE 2
412
DIRECTORIO
1A. SOLICITUD DE INICIAR TRANSACCIÓN
1B. RESPUESTA A INICIAR TRANSACCIÓN (TRANSID1)
2A. LEER ENTRADA (SOLICITUD DE DE BÚSQUEDA DE LDAP) [TRANSID1]
2B. OBTENER DATOS (ENTRADA DE RESULTADO DE BUSQUEDA DE LDAP) / BUSQUEDA COMPLETADA) [TRANSID1]
3A. LEER ENTRADA (SOLICITUD DE BÚSQUEDA DE LDAP)
3B. OBTENER DATOS (ENTRADA DE RESULTADO DE BUSQUEDA DE LDAP) / BUSQUEDA COMPLETADA)
4A. ACTUALIZAR DATOS EN ESTA ENTRADA (SOLICITUD DE MODIFICAR DE LDAP)
4B. ERROR (RESPUESTA A MODIFICAR DE LDAP [SIN ÉXITO)]
ESTE INTENTO DE ACTUALIZACIÓN SE HA RECHAZADO
5A. ACTUALIZAR DATOS EN ESTA ENTRADA (SOLICITUD DE MODIFICAR DE LDAP)
5B OK (RESPUESTA A MODIFICAR DE LDAP [RESULTADO: ÉXITO)]
6A. SOLICITUD DE DETENER TRANSACCIÓN [TRANSID1]
6B. RESPUESTA A DETENER TRANSACCIÓN [TRANSID1]
702
702
702
702''
702'
702'

FIG. 7

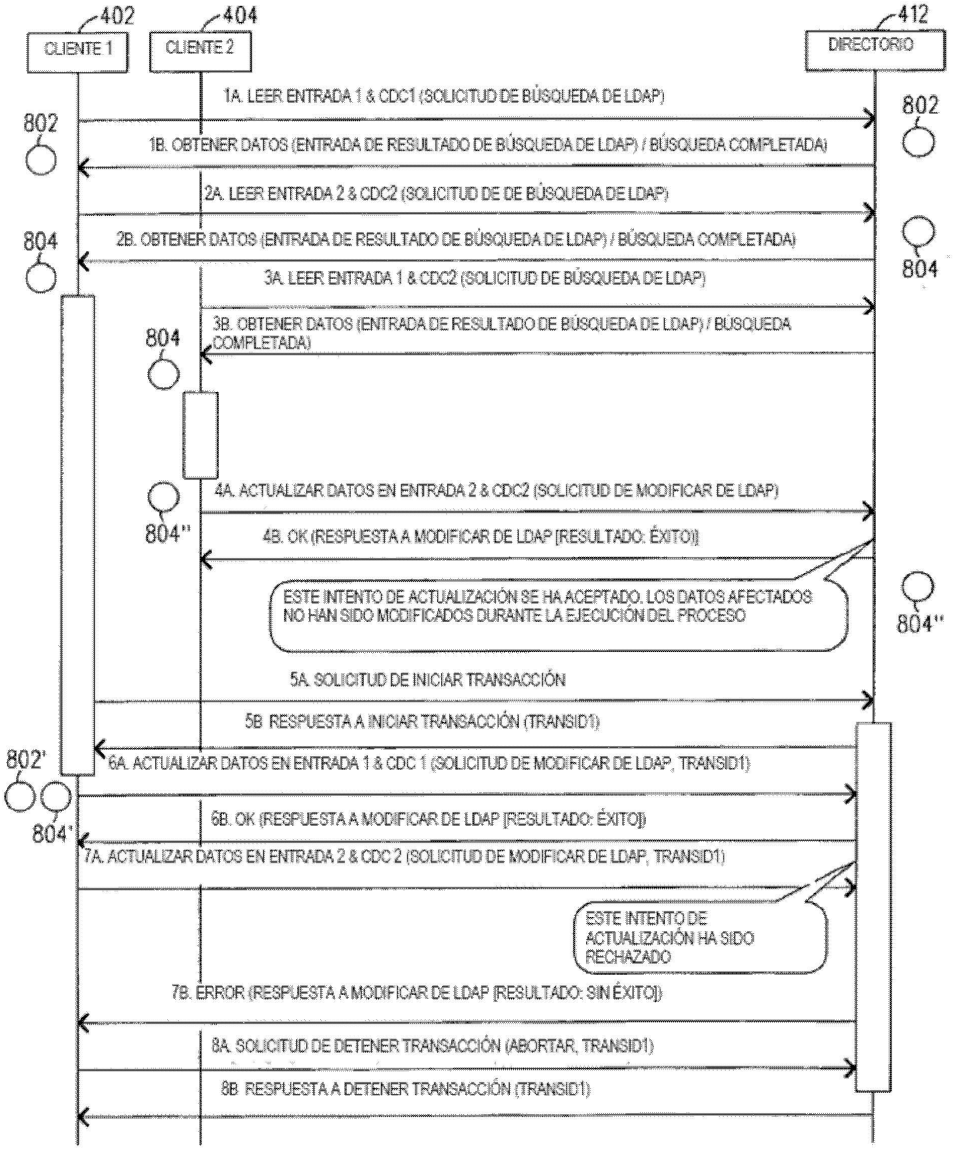

402
CLIENTE 1
404
CLIENTE 2
412
DIRECTORIO
1A. LEER ENTRADA 1 & CDC1 (SOLICITUD DE BÚSQUEDA DE LDAP)
802
1B. OBTENER DATOS (ENTRADA DE RESULTADO DE BÚSQUEDA DE LDAP) / BÚSQUEDA COMPLETADA)
802
2A. LEER ENTRADA 2 & CDC2 (SOLICITUD DE DE BÚSQUEDA DE LDAP)
2B. OBTENER DATOS (ENTRADA DE RESULTADO DE BÚSQUEDA DE LDAP) / BÚSQUEDA COMPLETADA)
804
804
3A. LEER ENTRADA 1 & CDC2 (SOLICITUD DE BÚSQUEDA DE LDAP)
3B. OBTENER DATOS (ENTRADA DE RESULTADO DE BÚSQUEDA DE LDAP) / BÚSQUEDA COMPLETADA)
804
4A. ACTUALIZAR DATOS EN ENTRADA 2 & CDC2 (SOLICITUD DE MODIFICAR DE LDAP)
804''
4B. OK (RESPUESTA A MODIFICAR DE LDAP [RESULTADO: ÉXITO])
ESTE INTENTO DE ACTUALIZACIÓN SE HA ACEPTADO. LOS DATOS AFECTADOS NO HAN SIDO MODIFICADOS DURANTE LA EJECUCIÓN DEL PROCESO
804''
5A. SOLICITUD DE INICIAR TRANSACCIÓN
5B RESPUESTA A INICIAR TRANSACCIÓN (TRANSID1)
802'
6A. ACTUALIZAR DATOS EN ENTRADA 1 & CDC 1 (SOLICITUD DE MODIFICAR DE LDAP, TRANSID1)
804'
6B. OK (RESPUESTA A MODIFICAR DE LDAP [RESULTADO: ÉXITO])
7A. ACTUALIZAR DATOS EN ENTRADA 2 & CDC 2 (SOLICITUD DE MODIFICAR DE LDAP, TRANSID1)
ESTE INTENTO DE ACTUALIZACIÓN HA SIDO RECHAZADO
7B. ERROR (RESPUESTA A MODIFICAR DE LDAP [RESULTADO: SIN ÉXITO])
8A. SOLICITUD DE DETENER TRANSACCIÓN (ABORTAR, TRANSID1)
8B RESPUESTA A DETENER TRANSACCIÓN (TRANSID1)

FIG. 8

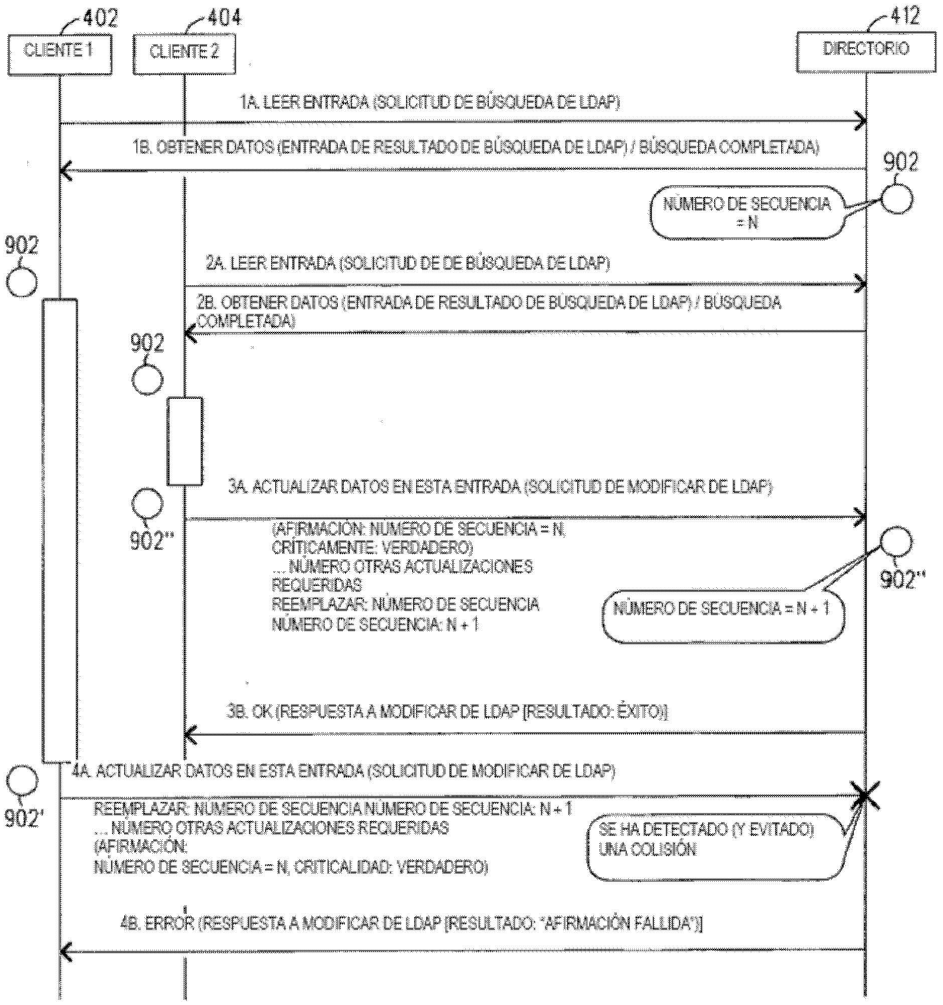

402
CLIENTE 1
404
CLIENTE 2
412
DIRECTORIO
1A. LEER ENTRADA (SOLICITUD DE BÚSQUEDA DE LDAP)
1B. OBTENER DATOS (ENTRADA DE RESULTADO DE BÚSQUEDA DE LDAP) / BÚSQUEDA COMPLETADA)
902
NÚMERO DE SECUENCIA = N
902
2A. LEER ENTRADA (SOLICITUD DE DE BÚSQUEDA DE LDAP)
2B. OBTENER DATOS (ENTRADA DE RESULTADO DE BÚSQUEDA DE LDAP) / BÚSQUEDA COMPLETADA)
902
3A. ACTUALIZAR DATOS EN ESTA ENTRADA (SOLICITUD DE MODIFICAR DE LDAP)
902''
(AFIRMACIÓN: NÚMERO DE SECUENCIA = N,
CRÍTICAMENTE: VERDADERO)
... NÚMERO OTRAS ACTUALIZACIONES REQUERIDAS
REEMPLAZAR: NÚMERO DE SECUENCIA
NÚMERO DE SECUENCIA: N + 1
NÚMERO DE SECUENCIA = N + 1
902''
3B. OK (RESPUESTA A MODIFICAR DE LDAP [RESULTADO: ÉXITO])
4A. ACTUALIZAR DATOS EN ESTA ENTRADA (SOLICITUD DE MODIFICAR DE LDAP)
902'
REEMPLAZAR: NÚMERO DE SECUENCIA NÚMERO DE SECUENCIA: N + 1
... NÚMERO OTRAS ACTUALIZACIONES REQUERIDAS
(AFIRMACIÓN:
NÚMERO DE SECUENCIA = N, CRITICALIDAD: VERDADERO)
SE HA DETECTADO (Y EVITADO) UNA COLISIÓN
4B. ERROR (RESPUESTA A MODIFICAR DE LDAP [RESULTADO: "AFIRMACIÓN FALLIDA"])

FIG. 9